```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBB          BBB   AAA       AAA   SSS           RRR         RRR       TTT        LLL
BBBBBBBBBBBB       AAA       AAA     SSSSSSSSS   RRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB       AAA       AAA     SSSSSSSSS   RRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB       AAA       AAA     SSSSSSSSS   RRRRRRRRRRR           TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR             TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBB          BBB   AAA       AAA           SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB       AAA       AAA   SSSSSSSSSSS   RRR         RRR       TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA       AAA   SSSSSSSSSSS   RRR         RRR       TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA       AAA   SSSSSSSSSSS   RRR         RRR       TTT        LLLLLLLLLLLLLLL
```

FILEID**BASCHANGE

```
BBBBBBBB      AAAAAA      SSSSSSSS    CCCCCCCC  HH      HH    AAAAAA    NN      NN    GGGGGGGG  EEEEEEEEEE
BBBBBBBB      AAAAAA      SSSSSSSS    CCCCCCCC  HH      HH    AAAAAA    NN      NN    GGGGGGGG  EEEEEEEEEE
BB      BB   AA      AA  SS          CC        HH      HH   AA      AA  NN      NN   GG        EE
BB      BB   AA      AA  SS          CC        HH      HH   AA      AA  NN      NN   GG        EE
BB      BB   AA      AA  SS          CC        HH      HH   AA      AA  NNNN    NN   GG        EE
BB      BB   AA      AA  SS          CC        HH      HH   AA      AA  NNNN    NN   GG        EE
BBBBBBBB     AA      AA    SSSSSS    CC        HHHHHHHHHH   AA      AA  NN  NN  NN   GG        EEEEEEE
BBBBBBBB     AA      AA    SSSSSS    CC        HHHHHHHHHH   AA      AA  NN  NN  NN   GG        EEEEEEE
BB      BB   AAAAAAAAAA          SS  CC        HH      HH   AAAAAAAAAA  NN    NNNN   GG  GGGGGG  EE
BB      BB   AAAAAAAAAA          SS  CC        HH      HH   AAAAAAAAAA  NN    NNNN   GG  GGGGGG  EE
BB      BB   AA      AA          SS  CC        HH      HH   AA      AA  NN      NN   GG      GG  EE
BB      BB   AA      AA          SS  CC        HH      HH   AA      AA  NN      NN   GG      GG  EE
BBBBBBBB     AA      AA  SSSSSSSS    CCCCCCCC  HH      HH   AA      AA  NN      NN     GGGGGG    EEEEEEEEEE
BBBBBBBB     AA      AA  SSSSSSSS    CCCCCCCC  HH      HH   AA      AA  NN      NN     GGGGGG    EEEEEEEEEE

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE BAS$CHANGE (                                  !
;     2          0002  0                     IDENT = '1-021'                 ! File: BASCHANGE.B32 EDIT: DG1021
;     3          0003  0                     ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************
;     7          0007  1 !*                                                                        *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                  *
;    11          0011  1 !*                                                                        *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    17          0017  1 !*  TRANSFERRED.                                                           *
;    18          0018  1 !*                                                                        *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    21          0021  1 !*  CORPORATION.                                                           *
;    22          0022  1 !*                                                                        *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !**************************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:  BASIC-PLUS-2 Miscellaneous
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This module contains routines which change a character string
;    36          0036  1 !       to a list of numbers and vice-versa.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:  VAX-11 User Mode
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: 20-FEB-1979
;    41          0041  1 !
;    42          0042  1 ! MODIFIED BY:
;    43          0043  1 !
;    44          0044  1 ! 1-001 - Original.  JBS 20-FEB-1979
;    45          0045  1 ! 1-002 - Track changes in the virtual array support code.  JBS 03-APR-1979
;    46          0046  1 ! 1-003 - Continue to track changes in the virtual array support
;    47          0047  1 !         code.  JBS 04-APR-1979
;    48          0048  1 ! 1-004 - Change OTS$S and LIB$S to STR$.  JBS 21-MAY-1979
;    49          0049  1 ! 1-005 - Change the index parameters to BAS$FETCH_BFA and BAS$STORE_BFA
;    50          0050  1 !         from by reference to by value.  JBS 01-JUN-1979
;    51          0051  1 ! 1-006 - Use BASLNK.  JBS 26-JUN-1979
;    52          0052  1 ! 1-007 - Change call to STR$COPY.  JBS 16-JUL-1979
;    53          0053  1 ! 1-008 - BAS$CHANGE_S_NA must apply the double precision scale
;    54          0054  1 !         to double precision arrays, and BAS$CHANGE_NA_S must
;    55          0055  1 !         descale before converting to a string.  PLL 22-May-1981
;    56          0056  1 ! 1-009 - BAS$CHANGE_S_NA was erroneously calling BAS$FETCH_BFA to store
;    57          0057  1 !         a value in a 2 dim. array.
```

```
;    58          0058  1 !           BAS$CHANGE_NA_S was not freeing it's dynamic string.
;    59          0059  1 !           Add support for new data types and dynamically mapped arrays.
;    60          0060  1 !           PLL 3-Mar-1982
;    61          0061  1 ! 1-010 - Add support for decimal arrays.  PLL 15-Mar-1982
;    62          0062  1 ! 1-011 - Correct arguments in CVTPL, CVTLP.  PLL 14-Apr-1982
;    63          0063  1 ! 1-012 - CVTPL should set scale to 0 and let FETCH_BFA do the scaling.
;    64          0064  1 !           PLL 16-Apr-82
;    65          0065  1 ! 1-013 - Clean up comments, etc from last edit.  PLL 21-Apr-82
;    66          0066  1 ! 1-014 - Add support for multiply dimensioned arrays.  PLL 24-May-82
;    67          0067  1 ! 1-015 - Fix bug in changing from string to integer arrays.  PLL 9-Jul-1982
;    68          0068  1 ! 1-016 - Fix bug in changing from integer to string.  PLL 26-Jul-1982
;    69          0069  1 ! 1-017 - Changing a string to a byte or word array does not store the value
;    70          0070  1 !           in the proper location.  Fix STORE.  PLL 13-Sep-1982
;    71          0071  1 ! 1-018 - Fix code which calculates the length for a virtual packed decimal
;    72          0072  1 !           array element.  (Must be power of 2.)  Also correct conversion
;    73          0073  1 !           of long to packed and vice versa.  Long must be converted to 10
;    74          0074  1 !           digit packed with 0 scale, and then to desired length and scale.
;    75          0075  1 !           While fixing miscellaneous bugs, also add some code to make
;    76          0076  1 !           dynamically mapped arrays work properly.  PLL 22-Sep-1982
;    77          0077  1 ! 1-019 - remove restriction of 255-byte destination character strings.
;    78          0078  1 !           dynamically allocate destination string based on length needed.
;    79          0079  1 !           MDL 14-Jun-1983
;    80          0080  1 ! 1-020 - Fixed: 1. if the string is longer than the numeric array, element 0
;    81          0081  1 !           of the numeric array contains the string's length.
;    82          0082  1 !                 2. if string length > 255 and CHANGEing to byte array,
;    83          0083  1 !           integer error is signalled.  DG 4-Jan-1984
;    84          0084  1 ! 1-021 - Dynamic remapped decimal arrays no longer get "Data type error".
;    85          0085  1 !           At the same time, fixed the array length calculations for data
;    86          0086  1 !           types longer than 1 longword.  DG 9-Jan-1984
;    87          0087  1 !--
;    88          0088  1
;    89          0089  1 !<BLF/PAGE>
```

```
;    91          0090  1 !
;    92          0091  1 ! SWITCHES:
;    93          0092  1 !
;    94          0093  1
;    95          0094  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    96          0095  1
;    97          0096  1 !
;    98          0097  1 ! LINKAGES:
;    99          0098  1 !
;   100          0099  1
;   101          0100  1 REQUIRE 'RTLIN:BASLNK';
;   102          0177  1 REQUIRE 'RTLIN:BASFRAME';                       ! BSF symbols
;   103          0380  1
;   104          0381  1 LINKAGE
;   105          0382  1     COPY_JSB = JSB (REGISTER = 0, REGISTER = 1) :
;   106          0383  1         NOTUSED (2,3,4,5,6,7,8,9,10,11);
;   107          0384  1
;   108          0385  1 !
;   109          0386  1 ! TABLE OF CONTENTS:
;   110          0387  1 !
;   111          0388  1
;   112          0389  1 FORWARD ROUTINE
;   113          0390  1     BAS$CHANGE_NA_S : NOVALUE,                  ! Change list to string
;   114          0391  1     BAS$CHANGE_S_NA : NOVALUE,                  ! Change string to list
;   115          0392  1     FETCH : NOVALUE,                            ! Fetch an array item
;   116          0393  1     STORE : NOVALUE;                            ! Store an array item
;   117          0394  1
;   118          0395  1 !
;   119          0396  1 ! INCLUDE FILES:
;   120          0397  1 !
;   121          0398  1
;   122          0399  1 REQUIRE 'RTLIN:RTLPSECT';                       ! Macros for defining psects
;   123          0494  1
;   124          0495  1 LIBRARY 'RTLSTARLE';                            ! System definitions
;   125          0496  1
;   126          0497  1 !
;   127          0498  1 ! MACROS:
;   128          0499  1 !
;   129          0500  1 !       NONE
;   130          0501  1 !
;   131          0502  1 ! EQUATED SYMBOLS:
;   132          0503  1 !
;   133          0504  1 !       NONE
;   134          0505  1 !
;   135          0506  1 ! PSECTS:
;   136          0507  1 !
;   137          0508  1 DECLARE_PSECTS (BAS);                           ! Declare psects for BAS$ facility
;   138          0509  1 !
;   139          0510  1 ! OWN STORAGE:
;   140          0511  1 !
;   141          0512  1 !       NONE
;   142          0513  1 !
;   143          0514  1 ! EXTERNAL REFERENCES:
;   144          0515  1 !
;   145          0516  1
;   146          0517  1 EXTERNAL ROUTINE
;   147          0518  1     BAS$$STOP : NOVALUE,                        ! signals fatal error
```

```
;  148          0519  1     BAS$SCALE_D_R1 : BAS$SCALE_LINK NOVALUE,    ! scale  a value
;  149          0520  1     BAS$DSCALE_D_R1 : BAS$SCALE_LINK NOVALUE,   ! descale a value
;  150          0521  1     BAS$$COPY_D_R1 : COPY_JSB NOVALUE,          ! copy a double number
;  151          0522  1     BAS$$VA_FETCH,                              ! fetch a virtual array element
;  152          0523  1     BAS$$VA_STORE,                              ! store a virtual array element
;  153          0524  1     STR$GETT_DX,                                ! allocate a string
;  154          0525  1     STR$FREET_DX,                               ! free a string
;  155          0526  1     STR$COPY_DX;                                ! copy a string
;  156          0527  1
;  157          0528  1 !+
;  158          0529  1 ! The following are the error codes used in this module.
;  159          0530  1 !-
;  160          0531  1
;  161          0532  1 EXTERNAL LITERAL
;  162          0533  1     BAS$K_MAXMEMEXC : UNSIGNED (8),             ! Maximum memory exceeded
;  163          0534  1     BAS$K_PROLOSSOR : UNSIGNED (8),             ! Program lost, sorry
;  164          0535  1     BAS$K_DATTYPERR : UNSIGNED (8),             ! Data type error
;  165          0536  1     BAS$K_ARGDONMAT : UNSIGNED (8),             ! Arguments don't match
;  166          0537  1     BAS$K_SUBOUTRAN : UNSIGNED (8),             ! Subscript out of range
;  167          0538  1     BAS$K_INTERR    : UNSIGNED (8),             ! Integer error
;  168          0539  1     BAS$K_NOTIMP : UNSIGNED (8);                ! Not implemented
;  169          0540  1
```

```
;  171     0541  1 GLOBAL ROUTINE BAS$CHANGE_NA_S (                      ! Change list to string
;  172     0542  1         LIST_DESC,                                     ! Address of array descriptor
;  173     0543  1         STR_RESULT                                     ! Result string
;  174     0544  1     ) : NOVALUE =
;  175     0545  1
;  176     0546  1 !++
;  177     0547  1 ! FUNCTIONAL DESCRIPTION:
;  178     0548  1 !
;  179     0549  1 !       Change the list of numbers to a string.  The first number is
;  180     0550  1 !       the length of the string.
;  181     0551  1 !
;  182     0552  1 ! FORMAL PARAMETERS:
;  183     0553  1 !
;  184     0554  1 !       LIST_DESC.rx.d  The list of numbers.  This may be word,
;  185     0555  1 !                       longword, floating or double.  It may be single-
;  186     0556  1 !                       or double-dimensioned.
;  187     0557  1 !       STR_RESULT.wt.d The descriptor for the string result.  It may
;  188     0558  1 !                       be dynamic or static.
;  189     0559  1 !
;  190     0560  1 ! IMPLICIT INPUTS:
;  191     0561  1 !
;  192     0562  1 !       NONE
;  193     0563  1 !
;  194     0564  1 ! IMPLICIT OUTPUTS:
;  195     0565  1 !
;  196     0566  1 !       NONE
;  197     0567  1 !
;  198     0568  1 ! ROUTINE VALUE:
;  199     0569  1 ! COMPLETION CODES:
;  200     0570  1 !
;  201     0571  1 !       NONE
;  202     0572  1 !
;  203     0573  1 ! SIDE EFFECTS:
;  204     0574  1 !
;  205     0575  1 !       NONE
;  206     0576  1 !
;  207     0577  1 !--
;  208     0578  1
;  209     0579  2     BEGIN
;  210     0580  2
;  211     0581  2 !+
;  212     0582  2 ! The FETCH routine will copy all numeric elements from LIST_DESC
;  213     0583  2 ! into the string buffer.
;  214     0584  2 !-
;  215     0585  2     FETCH (.LIST_DESC, .STR_RESULT);
;  216     0586  2
;  217     0587  2     RETURN;
;  218     0588  1     END;                                               ! end of BAS$CHANGE_NA_S


                                                        .TITLE  BAS$CHANGE
                                                        .IDENT  \1-021\

                                                        .EXTRN  BAS$$STOP, BAS$SCALE_D_R1
                                                        .EXTRN  BAS$DSCALE_D_R1
                                                        .EXTRN  BAS$$COPY_D_R1, BAS$$VA_FETCH
                                                        .EXTRN  BAS$$VA_STORE, STR$GET1_DX
```

```
                                                        .EXTRN  STR$FREE1_DX, STR$COPY_DX
                                                        .EXTRN  BAS$K_MAXMEMEXC
                                                        .EXTRN  BAS$K_PROLOSSOR
                                                        .EXTRN  BAS$K_DATTYPERR
                                                        .EXTRN  BAS$K_ARGDONMAT
                                                        .EXTRN  BAS$K_SUBOUTRAN
                                                        .EXTRN  BAS$K_INTERR, BAS$K_NOTIMP

                                                        .PSECT  _BAS$CODE,NOWRT,  SHR,  PIC,2

                                          0000 00000    .ENTRY  BAS$CHANGE_NA_S, Save nothing           ; 0541
                          7E        04    AC  7D 00002  MOVQ    LIST_DESC, -(SP)                        ; 0585
                0000V     CF              02  FB 00006  CALLS   #2, FETCH                               ;
                                          04 0000B      RET                                             ; 0588

; Routine Size:  12 bytes,    Routine Base:  _BAS$CODE + 0000

;  219          0589  1
```

```
;   221          0590  1 GLOBAL ROUTINE BAS$CHANGE_S_NA (                      ! Change string to list
;   222          0591  1         STR_DESC,                                     ! String to change
;   223          0592  1         LIST_RESULT                                   ! Where to put result
;   224          0593  1     ) : NOVALUE =
;   225          0594  1
;   226          0595  1 !++
;   227          0596  1 ! FUNCTIONAL DESCRIPTION:
;   228          0597  1 !
;   229          0598  1 !       Change the string to a list of numbers.  The first number is
;   230          0599  1 !       the length of the string.
;   231          0600  1 !
;   232          0601  1 ! FORMAL PARAMETERS:
;   233          0602  1 !
;   234          0603  1 !       STR_DESC.rt.d   The string to be converted to numbers.
;   235          0604  1 !       LIST_RESULT.wx.a The array of numbers to store in.  The type
;   236          0605  1 !                        may be word, longword, floating or double.
;   237          0606  1 !                        it may have one or two dimensions.
;   238          0607  1 !
;   239          0608  1 ! IMPLICIT INPUTS:
;   240          0609  1 !
;   241          0610  1 !       NONE
;   242          0611  1 !
;   243          0612  1 ! IMPLICIT OUTPUTS:
;   244          0613  1 !
;   245          0614  1 !       NONE
;   246          0615  1 !
;   247          0616  1 ! ROUTINE VALUE:
;   248          0617  1 ! COMPLETION CODES:
;   249          0618  1 !
;   250          0619  1 !       NONE
;   251          0620  1 !
;   252          0621  1 ! SIDE EFFECTS:
;   253          0622  1 !
;   254          0623  1 !       NONE
;   255          0624  1 !
;   256          0625  1 !--
;   257          0626  1
;   258          0627  2     BEGIN
;   259          0628  2
;   260          0629  2     MAP
;   261          0630  2         STR_DESC : REF BLOCK [8, BYTE];
;   262          0631  2
;   263          0632  2 !+
;   264          0633  2 ! Copy each character of the string to an element of the array.
;   265          0634  2 !-
;   266          0635  2     STORE (.STR_DESC, .LIST_RESULT);
;   267          0636  2     RETURN;
;   268          0637  1     END;                                              ! end of BAS$CHANGE_S_NA


                                0000 00000          .ENTRY  BAS$CHANGE_S_NA, Save nothing                   ; 0590
                         7E     04   AC  7D 00002   MOVQ    STR_DESC, -(SP)                                 ; 0635
                0000V    CF           02  FB 00006   CALLS   #2, STORE                                       ;
                                        04 0000B    RET                                                     ; 0637
```

```
; Routine Size:  12 bytes,    Routine Base:  _BAS$CODE + 000C

;  269          0638  1
```

```
;  271     0639  1 ROUTINE FETCH (                                  ! Fetch array values
;  272     0640  1         DESCRIP,                                  ! Array descriptor
;  273     0641  1         STR_DESC                                  ! Where to store values
;  274     0642  1     ) : NOVALUE =
;  275     0643  1
;  276     0644  1 !++
;  277     0645  1 ! FUNCTIONAL DESCRIPTION:
;  278     0646  1 !
;  279     0647  1 !       Fetch array values from an array or virtual array.  The array will
;  280     0648  1 !       always be numeric.  The values are changed to a string.
;  281     0649  1 !
;  282     0650  1 ! FORMAL PARAMETERS:
;  283     0651  1 !
;  284     0652  1 !       DESCRIP.rx.da   The descriptor of the array or virtual array
;  285     0653  1 !       STR_DESC.wx.dx  The string buffer to hold the values
;  286     0654  1 !
;  287     0655  1 ! IMPLICIT INPUTS:
;  288     0656  1 !
;  289     0657  1 !       NONE
;  290     0658  1 !
;  291     0659  1 ! IMPLICIT OUTPUTS:
;  292     0660  1 !
;  293     0661  1 !       NONE
;  294     0662  1 !
;  295     0663  1 ! ROUTINE VALUE:
;  296     0664  1 ! COMPLETION CODES:
;  297     0665  1 !
;  298     0666  1 !       NONE
;  299     0667  1 !
;  300     0668  1 ! SIDE EFFECTS:
;  301     0669  1 !
;  302     0670  1 !       Signals if an error is encountered.
;  303     0671  1 !
;  304     0672  1 !--
;  305     0673  1
;  306     0674  2     BEGIN
;  307     0675  2
;  308     0676  2     GLOBAL REGISTER
;  309     0677  2         BSF$A_MAJOR_STG = 11,
;  310     0678  2         BSF$A_MINOR_STG = 10,
;  311     0679  2         BSF$A_TEMP_STG = 9;
;  312     0680  2
;  313     0681  2     BUILTIN
;  314     0682  2         ASHP,
;  315     0683  2         CVTFL,
;  316     0684  2         CVTDL,
;  317     0685  2         CVTGL,
;  318     0686  2         CVTHL,
;  319     0687  2         CVTPL;
;  320     0688  2
;  321     0689  2     LOCAL
;  322     0690  2         TEMP_STR_DESC : BLOCK [8, BYTE],
;  323     0691  2         STR_STATUS,
;  324     0692  2         ARRAY_LEN,
;  325     0693  2         INDEX_VALUE,
;  326     0694  2         VALUE_LOCATION,
;  327     0695  2         MULTIPLIERS : REF VECTOR,
```

```
;  328          0696  2          BOUNDS : REF VECTOR,
;  329          0697  2          LOW_INDEX,
;  330          0698  2          HIGH_INDEX,
;  331          0699  2          INDEX_INCR,
;  332          0700  2          INDEX_NUMBER,
;  333          0701  2          VALUE_DESCR : BLOCK [12, BYTE],
;  334          0702  2          LENGTH,
;  335          0703  2          STR_BUF : REF VECTOR [, BYTE],
;  336          0704  2          STR_BUF_LONG,
;  337          0705  2          TEMP_LEN : VECTOR [4],
;  338          0706  2          TEMP_BUF : VECTOR [4];
;  339          0707  2
;  340          0708  2      MAP
;  341          0709  2          DESCRIP : REF BLOCK [8, BYTE],
;  342          0710  2          STR_DESC : REF BLOCK [8, BYTE];
;  343          0711  2
;  344          0712  2  !+
;  345          0713  2  ! The coefficients and bounds must be present.
;  346          0714  2  !-
;  347          0715  2
;  348          0716  2      IF ( NOT (.DESCRIP [DSC$V_FL_COEFF] AND .DESCRIP [DSC$V_FL_BOUNDS])) THEN BAS$$STOP (BAS$K_ARGDONMAT);
;  349          0717  2
;  350          0718  2      MULTIPLIERS = DESCRIP [DSC$L_M1];
;  351          0719  2      BOUNDS = DESCRIP [DSC$L_M1] + (%UPVAL*.DESCRIP [DSC$B_DIMCT]);
;  352          0720  2  !+
;  353          0721  2  ! Compute the lower and upper index numbers based on how the array
;  354          0722  2  ! is stored.
;  355          0723  2  !-
;  356          0724  2
;  357          0725  2      IF (.DESCRIP [DSC$V_FL_COLUMN])
;  358          0726  2      THEN
;  359          0727  3          BEGIN
;  360          0728  3          LOW_INDEX = .DESCRIP [DSC$B_DIMCT];
;  361          0729  3          HIGH_INDEX = 1;
;  362          0730  3          INDEX_INCR = -1;
;  363          0731  3          END
;  364          0732  2      ELSE
;  365          0733  3          BEGIN
;  366          0734  3          LOW_INDEX = 1;
;  367          0735  3          HIGH_INDEX = .DESCRIP [DSC$B_DIMCT];
;  368          0736  3          INDEX_INCR = 1;
;  369          0737  3          END;
;  370          0738  2
;  371          0739  2  !+
;  372          0740  2  ! If this is a decimal array, the length is the number of 4 bit digits
;  373          0741  2  ! (not including the sign).  Convert this to the number of bytes.
;  374          0742  2  ! Decimal virtual arrays and record virtual arrays are stored with
;  375          0743  2  ! a length that is a multiple of 2 - check for that here also.
;  376          0744  2  !-
;  377          0745  2      CASE .DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  378          0746  2          SET
;  379          0747  2
;  380          0748  3          [DSC$K_DTYPE_P]:                    ! decimal
;  381          0749  3              BEGIN
;  382          0750  3              LENGTH = (.DESCRIP [DSC$W_LENGTH]/2) + 1;
;  383          0751  3              IF .DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA
;  384          0752  3              THEN
```

```
;  385          0753  4                  BEGIN
;  386          0754  4
;  387          0755  5                  LENGTH = ( INCR I FROM 1 TO 9 BY 1 DO
;  388          0756  6                                  IF .LENGTH LSS (1 ^ .I)
;  389          0757  4                                  THEN EXITLOOP (1 ^ .I) );
;  390          0758  3                  END;
;  391          0759  2              END;
;  392          0760  2
;  393          0761  2          [INRANGE,OUTRANGE]:
;  394          0762  2              LENGTH = .DESCRIP [DSC$W_LENGTH];
;  395          0763  2          TES;
;  396          0764  2
;  397          0765  2      !+
;  398          0766  2      ! The number of elements in the array is stored in element 0.
;  399          0767  2      !-
;  400          0768  2
;  401          0769  2      IF .DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA
;  402          0770  2      THEN
;  403          0771  3          BEGIN
;  404          0772  3          IF .DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  405          0773  3          THEN
;  406          0774  4              BEGIN
;  407          0775  4              LOCAL
;  408          0776  4                  TEMP_DSC : BLOCK [12, BYTE];
;  409          0777  4              TEMP_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_P;
;  410          0778  4              TEMP_DSC [DSC$B_CLASS] = DSC$K_CLASS_SD;
;  411          0779  4              TEMP_DSC [DSC$W_LENGTH] = .DESCRIP [DSC$W_LENGTH];
;  412          0780  4              TEMP_DSC [DSC$A_POINTER] = TEMP_LEN [0];
;  413          0781  4              TEMP_DSC [DSC$B_SCALE] = .DESCRIP [DSC$B_SCALE];
;  414          0782  4              BAS$$VA_FETCH (.DESCRIP, 0, TEMP_DSC)
;  415          0783  4              END
;  416          0784  3          ELSE
;  417          0785  3              BAS$$VA_FETCH (.DESCRIP, 0, TEMP_LEN)
;  418          0786  3          END
;  419          0787  2      ELSE
;  420          0788  2          CASE .DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  421          0789  2          SET
;  422          0790  2              [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L, DSC$K_DTYPE_F] :
;  423          0791  2                  TEMP_LEN = .(.DESCRIP [DSC$A_POINTER]);
;  424          0792  2
;  425          0793  2              [DSC$K_DTYPE_D, DSC$K_DTYPE_G] :
;  426          0794  3                  BEGIN
;  427          0795  3
;  428          0796  3                  TEMP_LEN[0] = .(.DESCRIP [DSC$A_POINTER]);
;  429          0797  3                  TEMP_LEN[1] = .(.DESCRIP [DSC$A_POINTER] + 4);
;  430          0798  3
;  431          0799  2                  END;
;  432          0800  2
;  433          0801  2              [DSC$K_DTYPE_H] :
;  434          0802  3                  BEGIN
;  435          0803  3
;  436          0804  3                  TEMP_LEN[0] = .(.DESCRIP [DSC$A_POINTER]);
;  437          0805  3                  TEMP_LEN[1] = .(.DESCRIP [DSC$A_POINTER] + 4);
;  438          0806  3                  TEMP_LEN[2] = .(.DESCRIP [DSC$A_POINTER] + 8);
;  439          0807  3                  TEMP_LEN[3] = .(.DESCRIP [DSC$A_POINTER] + 12);
;  440          0808  3
;  441          0809  2                  END;
```

```
;  442          0810  2
;  443          0811  2              [DSC$K_DTYPE_P] :
;  444          0812  2                  CH$MOVE ((.DESCRIP [DSC$W_LENGTH]/2) + 1,
;  445          0813  2                           .DESCRIP [DSC$A_POINTER], TEMP_LEN);
;  446          0814  2
;  447          0815  2              [DSC$K_DTYPE_DSC] :
;  448          0816  2                  ;
;  449          0817  2
;  450          0818  2              [INRANGE,OUTRANGE] :
;  451          0819  2                  BAS$$STOP (BAS$K_DATTYPERR);
;  452          0820  2          TES;
;  453          0821  2
;  454          0822  2      CASE .DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  455          0823  2      SET
;  456          0824  2          [DSC$K_DTYPE_B] :
;  457          0825  2              ARRAY_LEN = .BLOCK [TEMP_LEN, 0, 0, %BPUNIT, 1];
;  458          0826  2
;  459          0827  2          [DSC$K_DTYPE_W] :
;  460          0828  2              ARRAY_LEN = .BLOCK [TEMP_LEN, 0, 0, %BPVAL/2, 1];
;  461          0829  2
;  462          0830  2          [DSC$K_DTYPE_L] :
;  463          0831  2              ARRAY_LEN = .TEMP_LEN;
;  464          0832  2
;  465          0833  2          [DSC$K_DTYPE_F] :
;  466          0834  2              CVTFL (TEMP_LEN, ARRAY_LEN);
;  467          0835  2
;  468          0836  2          [DSC$K_DTYPE_D] :
;  469          0837  3              BEGIN
;  470          0838  3              !+
;  471          0839  3              ! A double value must be de-scaled before it can be used.
;  472          0840  3              !-
;  473          0841  3              LOCAL
;  474          0842  3                  TEMP_DBL : VECTOR [2];
;  475          0843  3              REGISTER
;  476          0844  3                  R0 = 0,
;  477          0845  3                  R1 = 1;
;  478          0846  3              BAS$$COPY_D_R1 (TEMP_LEN, TEMP_DBL [0]);
;  479          0847  3              BAS$DSCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  480          0848  3              TEMP_DBL [0] = .R0;
;  481          0849  3              TEMP_DBL [1] = .R1;
;  482          0850  3              CVTD[ (TEMP_DBL [0], ARRAY_LEN);
;  483          0851  3              END;
;  484          0852  2
;  485          0853  2          [DSC$K_DTYPE_G] :
;  486          0854  2              CVTGL (TEMP_LEN, ARRAY_LEN);
;  487          0855  2
;  488          0856  2          [DSC$K_DTYPE_H] :
;  489          0857  2              CVTHL (TEMP_LEN, ARRAY_LEN);
;  490          0858  2
;  491          0859  2          [DSC$K_DTYPE_P] :
;  492          0860  2              BEGIN
;  493          0861  3                  LOCAL
;  494          0862  3                      TEMP_P : VECTOR [6,BYTE];
;  495          0863  3              ASHP (DESCRIP [DSC$B_SCALE], DESCRIP [DSC$W_LENGTH],
;  496          0864  3                  TEMP_LEN [0], %REF(0), %REF(10), TEMP_P [0]);
;  497          0865  3              CVTPL (%REF(10), TEMP_P, ARRAY_LEN);
;  498          0866  2              END;
```

```
;  499          0867  2
;  500          0868  2          [DSC$K_DTYPE_DSC] :                     ! dynamically mapped array
;  501          0869  2              BEGIN
;  502          0870  2              LOCAL
;  503          0871  2                  ELEM_DESC : REF BLOCK [8,BYTE];
;  504          0872  2
;  505          0873  2              IF .DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA
;  506          0874  2              THEN
;  507          0875  2                  BAS$$STOP (BAS$K_NOTIMP);       ! no virtual dyn mapped arrays
;  508          0876  2
;  509          0877  2              ELEM_DESC = .DESCRIP [DSC$A_POINTER];
;  510          0878  2              CASE .ELEM_DESC [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  511          0879  2              SET
;  512          0880  2                  [DSC$K_DTYPE_B] :
;  513          0881  2                      ARRAY_LEN =
;  514          0882  2                          .BLOCK [.ELEM_DESC [DSC$A_POINTER], 0, 0, %BPUNIT, 1];
;  515          0883  2
;  516          0884  2                  [DSC$K_DTYPE_W] :
;  517          0885  2                      ARRAY_LEN =
;  518          0886  2                          .BLOCK [.ELEM_DESC [DSC$A_POINTER], 0, 0, %BPVAL/2, 1];
;  519          0887  2
;  520          0888  2                  [DSC$K_DTYPE_L] :
;  521          0889  2                      ARRAY_LEN = .(.ELEM_DESC [DSC$A_POINTER]);
;  522          0890  2
;  523          0891  2                  [DSC$K_DTYPE_F] :
;  524          0892  2                      CVTFL (.ELEM_DESC [DSC$A_POINTER], ARRAY_LEN);
;  525          0893  2
;  526          0894  2                  [DSC$K_DTYPE_D] :
;  527          0895  4                      BEGIN
;  528          0896  4                      LOCAL
;  529          0897  4                          TEMP_DBL : VECTOR [2];
;  530          0898  4                      REGISTER
;  531          0899  4                          R0 = 0,
;  532          0900  4                          R1 = 1;
;  533          0901  4                      BAS$$COPY_D_R1 (.ELEM_DESC [DSC$A_POINTER], TEMP_DBL [0]);
;  534          0902  4                      BAS$DSCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  535          0903  4                      TEMP_DBL [0] = .R0;
;  536          0904  4                      TEMP_DBL [1] = .R1;
;  537          0905  4                      CVTDL (TEMP_DBL [0], ARRAY_LEN);
;  538          0906  4                      END;
;  539          0907  2
;  540          0908  2                  [DSC$K_DTYPE_G] :
;  541          0909  2                      CVTGL  (.ELEM_DESC [DSC$A_POINTER], ARRAY_LEN);
;  542          0910  2
;  543          0911  2                  [DSC$K_DTYPE_H] :
;  544          0912  2                      CVTHL (.ELEM_DESC [DSC$A_POINTER], ARRAY_LEN);
;  545          0913  2
;  546          0914  2                  [DSC$K_DTYPE_P] :
;  547          0915  4                      BEGIN
;  548          0916  4                          LOCAL
;  549          0917  4                              TEMP_P : VECTOR [6,BYTE];
;  550          0918  4                      ASHP (ELEM_DESC [DSC$B_SCALE], ELEM_DESC [DSC$W_LENGTH],
;  551          0919  4                          .ELEM_DESC [DSC$A_POINTER], %REF(0), %REF(10),
;  552          0920  4                          TEMP_P [0]);
;  553          0921  4                      CVTPL (%REF(10), TEMP_P, ARRAY_LEN);
;  554          0922  3                      END;
;  555          0923  3
```

```
;  556          0924  3                  [INRANGE, OUTRANGE] :
;  557          0925  3                      BAS$$STOP (BAS$K_DATTYPERR);
;  558          0926  3
;  559          0927  3                  TES;
;  560          0928  3                  END;
;  561          0929  3
;  562          0930  3              [INRANGE, OUTRANGE] :
;  563          0931  3                  BAS$$STOP (BAS$K_DATTYPERR);
;  564          0932  3          TES;
;  565          0933  3
;  566          0934  3      !+
;  567          0935  3      ! Now that we know how long the array is, we can allocate a temporary string
;  568          0936  3      ! to CHANGE the array into.
;  569          0937  3      !-
;  570          0938  3          TEMP_STR_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
;  571          0939  3          TEMP_STR_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  572          0940  3          TEMP_STR_DESC [DSC$W_LENGTH] = 0;
;  573          0941  3          TEMP_STR_DESC [DSC$A_POINTER] = 0;
;  574          0942  3          STR_STATUS = STR$GET_T_DX (ARRAY_LEN, TEMP_STR_DESC);
;  575          0943  3          IF NOT .STR_STATUS
;  576          0944  3          THEN
;  577          0945  3              BAS$$STOP (.STR_STATUS);
;  578          0946  3          STR_BUF = .TEMP_STR_DESC [DSC$A_POINTER];
;  579          0947  3
;  580          0948  3      !+
;  581          0949  3      ! Compute linear index.  Note that all indicies will be zero except for one,
;  582          0950  3      ! since CHANGE operates only on row 0.  This code should accomodate FORTRAN
;  583          0951  3      ! arrays.
;  584          0952  3      !-
;  585          0953  3
;  586          0954  3
;  587          0955  3          INCR INDEX FROM 1 TO .ARRAY_LEN DO
;  588          0956  3          BEGIN
;  589          0957  3              INDEX_NUMBER = .LOW_INDEX - .INDEX_INCR;
;  590          0958  3              INDEX_VALUE = .INDEX;
;  591          0959  3              VALUE_LOCATION = 0;
;  592          0960  3
;  593          0961  3              WHILE ((INDEX_NUMBER = .INDEX_NUMBER + .INDEX_INCR) NEQ (.HIGH_INDEX + .INDEX_INCR)) DO
;  594          0962  4              BEGIN
;  595          0963  4              IF ((.INDEX_VALUE LSS .BOUNDS [(.INDEX_NUMBER - 1)*2])
;  596          0964  4                  OR (.INDEX_VALUE GTR .BOUNDS [((.INDEX_NUMBER - 1)*2) + 1]))
;  597          0965  4              THEN
;  598          0966  5                  BEGIN
;  599          0967  5                  STR$FREE1_DX (TEMP_STR_DESC);
;  600          0968  5                  BAS$$STOP (BAS$K_SUBOUTRAN);
;  601          0969  5                  END;
;  602          0970  4              VALUE_LOCATION = (.VALUE_LOCATION*.MULTIPLIERS [.INDEX_NUMBER - 1]) + .INDEX_VALUE;
;  603          0971  4              INDEX_VALUE = 0;                        ! all indicies except 1st are zero
;  604          0972  4              END;
;  605          0973  3
;  606          0974  3          VALUE_LOCATION = (.VALUE_LOCATION*.LENGTH) + .DESCRIP [DSC$A_A0];
;  607          0975  3
;  608          0976  3      !+
;  609          0977  3      ! Build a descriptor pointing to the value cell in the array.  If this
;  610          0978  3      ! is an array of descriptors, the descriptor is copied, otherwise it
;  611          0979  3      ! is constructed.
;  612          0980  3      !-
```

```
;  613          0981  3
;  614          0982  4      IF (.DESCRIP [DSC$B_DTYPE] EQLU DSC$K_DTYPE_DSC)
;  615          0983  3      THEN
;  616          0984  4          BEGIN
;  617          0985  4
;  618          0986  4          MAP
;  619          0987  4              VALUE_LOCATION : REF BLOCK [8, BYTE];
;  620          0988  4
;  621          0989  4          VALUE_DESCR [DSC$W_LENGTH] = .VALUE_LOCATION [DSC$W_LENGTH];
;  622          0990  4          VALUE_DESCR [DSC$B_DTYPE] = .VALUE_LOCATION [DSC$B_DTYPE];
;  623          0991  5          VALUE_DESCR [DSC$B_CLASS] = (IF (.VALUE_LOCATION [DSC$B_CLASS] EQLU DSC$K_CLASS_D) THEN DSC$K_CLASS_
;  624          0992  4              ELSE .VALUE_LOCATION [DSC$B_CLASS]);
;  625          0993  4          VALUE_DESCR [DSC$A_POINTER] = .VALUE_LOCATION [DSC$A_POINTER];
;  626          0994  4          IF .VALUE_DESCR [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  627          0995  4          THEN
;  628          0996  5              BEGIN
;  629          0997  5              MAP
;  630          0998  5                  VALUE_LOCATION : REF BLOCK [12,BYTE];
;  631          0999  5              VALUE_DESCR [DSC$B_SCALE] = .VALUE_LOCATION [DSC$B_SCALE];
;  632          1000  4              END;
;  633          1001  4          END
;  634          1002  3      ELSE
;  635          1003  4          BEGIN
;  636          1004  4          VALUE_DESCR [DSC$W_LENGTH] = .DESCRIP [DSC$W_LENGTH];
;  637          1005  4          VALUE_DESCR [DSC$B_DTYPE] = .DESCRIP [DSC$B_DTYPE];
;  638          1006  4          VALUE_DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  639          1007  4          VALUE_DESCR [DSC$A_POINTER] = .VALUE_LOCATION;
;  640          1008  4          IF .VALUE_DESCR [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  641          1009  4          THEN
;  642          1010  5              BEGIN
;  643          1011  5              MAP
;  644          1012  5                  DESCRIP : REF BLOCK [12,BYTE];
;  645          1013  5              VALUE_DESCR [DSC$B_SCALE] = .DESCRIP [DSC$B_SCALE];
;  646          1014  4              END;
;  647          1015  3          END;
;  648          1016  3
;  649          1017  3      !+
;  650          1018  3      ! Special handling if this is a virtual array.
;  651          1019  3      !-
;  652          1020  3
;  653          1021  4          IF (.DESCRIP [DSC$B_CLASS] EQLU DSC$K_CLASS_BFA)
;  654          1022  3          THEN
;  655          1023  4              BEGIN
;  656          1024  4
;  657          1025  5              IF (.DESCRIP [DSC$B_DTYPE] EQLU DSC$K_DTYPE_DSC)
;  658          1026  4              THEN
;  659          1027  5                  BEGIN
;  660          1028  5                  STR$FREE1_DX (TEMP_STR_DESC);
;  661          1029  5                  BAS$$STOP (BAS$K_NOTIMP);
;  662          1030  4                  END;
;  663          1031  4
;  664          1032  4              IF .DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  665          1033  4              THEN
;  666          1034  5                  BEGIN
;  667          1035  5                  LOCAL
;  668          1036  5                      TEMP_DSC : BLOCK [12, BYTE];
;  669          1037  5                  TEMP_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_P;
```

```
;  670          1038  5              TEMP_DSC [DSC$B_CLASS] = DSC$K_CLASS_SD;
;  671          1039  5              TEMP_DSC [DSC$W_LENGTH] = .DESCRIP [DSC$W_LENGTH];
;  672          1040  5              TEMP_DSC [DSC$A_POINTER] = TEMP_BUF [0];
;  673          1041  5              TEMP_DSC [DSC$B_SCALE] = .DESCRIP [DSC$B_SCALE];
;  674          1042  5              BAS$$VA_FETCH (.DESCRIP, .VALUE_LOCATION, TEMP_DSC)
;  675          1043  5              END
;  676          1044  4          ELSE
;  677          1045  4              BAS$$VA_FETCH (.DESCRIP, .VALUE_LOCATION, TEMP_BUF [0]);
;  678          1046  4
;  679          1047  4          VALUE_DESCR [DSC$A_POINTER] = TEMP_BUF [0];
;  680          1048  4
;  681          1049  4          END
;  682          1050  3      ELSE
;  683          1051  3
;  684          1052  4          IF (.DESCRIP [DSC$B_CLASS] NEQU DSC$K_CLASS_A)
;  685          1053  3          THEN
;  686          1054  4              BEGIN
;  687          1055  4              STR$FREE1_DX (TEMP_STR_DESC);
;  688          1056  4              BAS$$STOP (BAS$K_NOTIMP);
;  689          1057  4              END;
;  690          1058  3
;  691          1059  3      !+
;  692          1060  3      ! Data is converted to longword (to use BUILTINs) and then to byte.
;  693          1061  3      !-
;  694          1062  3
;  695          1063  3          CASE .VALUE_DESCR [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  696          1064  3              SET
;  697          1065  3
;  698          1066  3              [DSC$K_DTYPE_B, DSC$K_DTYPE_W, DSC$K_DTYPE_L] :
;  699          1067  3                  STR_BUF [.INDEX - 1] = .(.VALUE_DESCR [DSC$A_POINTER]);
;  700          1068  3
;  701          1069  3              [DSC$K_DTYPE_F] :                   ! 32-bit floating point
;  702          1070  4                  BEGIN
;  703          1071  4                  CVTFL (.VALUE_DESCR [DSC$A_POINTER], STR_BUF_LONG);
;  704          1072  4                  STR_BUF [.INDEX - 1] = .STR_BUF_LONG;
;  705          1073  3                  END;
;  706          1074  3
;  707          1075  3              [DSC$K_DTYPE_D] :                   ! 64-bit double floating
;  708          1076  4                  BEGIN
;  709          1077  4                  !+
;  710          1078  4                  ! Double values may need to be de-scaled.
;  711          1079  4                  !-
;  712          1080  4                  LOCAL
;  713          1081  4                      TEMP_DBL : VECTOR [2];
;  714          1082  4                  REGISTER
;  715          1083  4                      R0 = 0,
;  716          1084  4                      R1 = 1;
;  717          1085  4                  BAS$$COPY_D_R1 (.VALUE_DESCR [DSC$A_POINTER], TEMP_DBL [0]);
;  718          1086  4                  BAS$DSCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  719          1087  4                  TEMP_DBL [0] = .R0;
;  720          1088  4                  TEMP_DBL [1] = .R1;
;  721          1089  4                  CVTDL (TEMP_DBL [0], STR_BUF_LONG);
;  722          1090  4                  STR_BUF [.INDEX - 1] = .STR_BUF_LONG;
;  723          1091  3                  END;
;  724          1092  3
;  725          1093  3              [DSC$K_DTYPE_G] :                   ! G floating
;  726          1094  4                  BEGIN
```

```
;  727     1095  4                CVTGL (.VALUE_DESCR [DSC$A_POINTER], STR_BUF_LONG);
;  728     1096  4                STR_BUF [.INDEX - 1] = .STR_BUF_LONG;
;  729     1097  3                END;
;  730     1098  3
;  731     1099  3            [DSC$K_DTYPE_H] :                   ! H floating
;  732     1100  4                BEGIN
;  733     1101  4                CVTHL (.VALUE_DESCR [DSC$A_POINTER], STR_BUF_LONG);
;  734     1102  4                STR_BUF [.INDEX - 1] = .STR_BUF_LONG;
;  735     1103  3                END;
;  736     1104  3
;  737     1105  3            [DSC$K_DTYPE_P] :                   ! decimal
;  738     1106  4                BEGIN
;  739     1107  4                LOCAL
;  740     1108  4                    TEMP_P : VECTOR [6,BYTE];
;  741     1109  4                ASHP (VALUE_DESCR [DSC$B_SCALE], VALUE_DESCR [DSC$W_LENGTH],
;  742     1110  4                    .VALUE_DESCR [DSC$A_POINTER], %REF(0), %REF(10),
;  743     1111  4                    TEMP_P);
;  744     1112  4                CVTPL (%REF(10), TEMP_P, STR_BUF_LONG);
;  745     1113  4                STR_BUF [.INDEX - 1] = .STR_BUF_LONG;
;  746     1114  3                END;
;  747     1115  3
;  748     1116  3            [DSC$K_DTYPE_DSC] :                 ! dynamically mapped array
;  749     1117  4                BEGIN
;  750     1118  4
;  751     1119  4                IF .DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA
;  752     1120  4                THEN
;  753     1121  5                    BEGIN
;  754     1122  5                    STR$FREE1_DX (TEMP_STR_DESC);
;  755     1123  5                    BAS$$STOP (BAS$K_NOTIMP);   ! no virtual dyn mapped arrays
;  756     1124  4                    END;
;  757     1125  4
;  758     1126  4                CASE .VALUE_DESCR [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  759     1127  4                SET
;  760     1128  4                    [DSC$K_DTYPE_B] :
;  761     1129  4                        STR_BUF_LONG =
;  762     1130  4                            .BLOCK [.VALUE_DESCR [DSC$A_POINTER], 0, 0, %BPUNIT, 1];
;  763     1131  4
;  764     1132  4                    [DSC$K_DTYPE_W] :
;  765     1133  4                        STR_BUF_LONG =
;  766     1134  4                            .BLOCK [.VALUE_DESCR [DSC$A_POINTER], 0, 0, %BPVAL/2, 1];
;  767     1135  4
;  768     1136  4                    [DSC$K_DTYPE_L] :
;  769     1137  4                        STR_BUF_LONG = .(.VALUE_DESCR [DSC$A_POINTER]);
;  770     1138  4
;  771     1139  4                    [DSC$K_DTYPE_F] :
;  772     1140  4                        CVTFL (.VALUE_DESCR [DSC$A_POINTER], STR_BUF_LONG);
;  773     1141  4
;  774     1142  4                    [DSC$K_DTYPE_D] :
;  775     1143  5                        BEGIN
;  776     1144  5                        LOCAL
;  777     1145  5                            TEMP_DBL : VECTOR [2];
;  778     1146  5                        REGISTER
;  779     1147  5                            R0 = 0,
;  780     1148  5                            R1 = 1;
;  781     1149  5                        BAS$$COPY_D_R1 (.VALUE_DESCR [DSC$A_POINTER], TEMP_DBL [0]);
;  782     1150  5                        BAS$DSCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  783     1151  5                        TEMP_DBL [0] = .R0;
```

```
;  784          1152  5                    TEMP_DBL [1] = .R1;
;  785          1153  5                    CVTDL (TEMP_DBL [0], STR_BUF_LONG);
;  786          1154  4                    END;
;  787          1155  4
;  788          1156  4                [DSC$K_DTYPE_G] :
;  789          1157  4                    CVTGL (.VALUE_DESCR [DSC$A_POINTER], STR_BUF_LONG);
;  790          1158  4
;  791          1159  4                [DSC$K_DTYPE_H] :
;  792          1160  4                    CVTHL (.VALUE_DESCR [DSC$A_POINTER], STR_BUF_LONG);
;  793          1161  4
;  794          1162  4                [DSC$K_DTYPE_P] :                   ! decimal
;  795          1163  5                    BEGIN
;  796          1164  5                    LOCAL
;  797          1165  5                        TEMP_P : VECTOR [6,BYTE];
;  798          1166  5                    ASHP (VALUE_DESCR [DSC$B_SCALE], VALUE_DESCR [DSC$W_LENGTH],
;  799          1167  5                        .VALUE_DESCR [DSC$A_POINTER], %REF(0), %REF(10),
;  800          1168  5                        TEMP_P);
;  801          1169  5                    CVTPL (%REF(10), TEMP_P, STR_BUF_LONG);
;  802          1170  5                    STR_BUF [.INDEX - 1] = .STR_BUF_LONG;
;  803          1171  4                    END;
;  804          1172  4
;  805          1173  4                [INRANGE, OUTRANGE] :
;  806          1174  5                    BEGIN
;  807          1175  5                    STR$FREE1_DX (TEMP_STR_DESC);
;  808          1176  5                    BAS$$STOP (BAS$K_DATTYPERR);
;  809          1177  4                    END;
;  810          1178  4
;  811          1179  4                TES;
;  812          1180  3                END;
;  813          1181  3
;  814          1182  3            [INRANGE, OUTRANGE] :
;  815          1183  4                BEGIN
;  816          1184  4                STR$FREE1_DX (TEMP_STR_DESC);
;  817          1185  4                BAS$$STOP (BAS$K_DATTYPERR);
;  818          1186  4                END;
;  819          1187  3
;  820          1188  3            TES;
;  821          1189  2
;  822          1190  2        END;                                        ! end of INCR loop
;  823          1191  2
;  824          1192  2        !+
;  825          1193  2        ! copy string back to caller
;  826          1194  2        !-
;  827          1195  2        STR$COPY_DX (.STR_DESC, TEMP_STR_DESC);
;  828          1196  2        !+
;  829          1197  2        ! free temporary string
;  830          1198  2        !-
;  831          1199  2        STR$FREE1_DX (TEMP_STR_DESC);
;  832          1200  2
;  833          1201  1        END;                                        ! end of FETCH
; INFO#250                   L1:0848
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250                   L1:0849
; Referenced REGISTER symbol R1 is probably not initialized
; INFO#250                   L1:0903
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250                   L1:0904
```

```
; Referenced REGISTER symbol R1 is probably not initialized
; INFO#250              L1:1087
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250              L1:1088
; Referenced REGISTER symbol R1 is probably not initialized
; INFO#250              L1:1151
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250              L1:1152
; Referenced REGISTER symbol R1 is probably not initialized


                                          0FFC 00000 FETCH:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 0639
                       5E        A0    AE  9E 00002          MOVAB   -96(SP), SP                               ;
           50      04  AC              0A  C1 00006          ADDL3   #10, DESCRIP, R0                          ; 0716
           09          60              06  E1 0000B          BBC     #6, (R0), 1$                              ;
           51      04  AC              0A  C1 0000F          ADDL3   #10, DESCRIP, R1                          ;
                                       61  95 00014          TSTB    (R1)                                      ;
                                       0B  19 00016          BLSS    2$                                        ;
                       7E       00G    8F  9A 00018 1$:      MOVZBL  #BAS$K_ARGDONMAT, -(SP)                   ;
              00000000G 00             01  FB 0001C          CALLS   #1, BAS$$STOP                             ;
      10   AE      04  AC              14  C1 00023 2$:      ADDL3   #20, DESCRIP, MULTIPLIERS                 ; 0718
           51      04  AC              0B  C1 00029          ADDL3   #11, DESCRIP, R1                          ; 0719
                       50              61  9A 0002E          MOVZBL  (R1), R0                                  ;
           51      04  AC              14  C1 00031          ADDL3   #20, DESCRIP, R1                          ;
                   08  AE            6140  DE 00036          MOVAL   (R1)[R0], BOUNDS                          ;
           51      04  AC              0A  C1 0003B          ADDL3   #10, DESCRIP, R1                          ; 0725
           0C          61              05  E1 00040          BBC     #5, (R1), 3$                              ;
                       57              50  D0 00044          MOVL    R0, LOW_INDEX                             ; 0728
                       56              01  D0 00047          MOVL    #1, HIGH_INDEX                            ; 0729
                   0C  AE              01  CE 0004A          MNEGL   #1, INDEX_INCR                            ; 0730
                                       0A  11 0004E          BRB     4$                                        ; 0725
                       57              01  D0 00050 3$:      MOVL    #1, LOW_INDEX                             ; 0734
                       56              50  D0 00053          MOVL    R0, HIGH_INDEX                            ; 0735
                   0C  AE              01  D0 00056          MOVL    #1, INDEX_INCR                            ; 0736
           53      04  AC              02  C1 0005A 4$:      ADDL3   #2, DESCRIP, R3                           ; 0745
           16          06              63  8F 0005F          CASEB   (R3), #6, #22                             ;
 002E      002E        002E          002E     00063 5$:      .WORD   6$-5$,-                                   ;
 002E      002E        002E          002E     0006B                  6$-5$,-                                   ;
 002E      002E        002E          002E     00073                  6$-5$,-                                   ;
 0037      002E        002E          002E     0007B                  6$-5$,-                                   ;
 002E      002E        002E          002E     00083                  6$-5$,-                                   ;
           002E        002E          002E     0008B                  6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     7$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
                                                                     6$-5$,-                                   ;
```

```
                                                                      6$-5$,-                              ;
                                                                      6$-5$,-                              ;
                                                                      6$-5$,-                              ;
                                                                      6$-5$,-                              ;
                                                                      6$-5$                                ;
                 52        04    BC   3C 00091 6$:     MOVZWL  @DESCRIP, R2                         ; 0762
                 58              52   D0 00095         MOVL    R2, LENGTH                           ;
                                 2F   11 00098         BRB     10$                                  ;
                 52        04    BC   3C 0009A 7$:     MOVZWL  @DESCRIP, R2                         ; 0750
        50       52              02   C7 0009E         DIVL3   #2, R2, R0                           ;
                 58        01    A0   9E 000A2         MOVAB   1(R0), LENGTH                        ;
        50  04   AC              03   C1 000A6         ADDL3   #3, DESCRIP, R0                      ; 0751
            BF   8F              60   91 000AB         CMPB    (R0), #191                           ;
                                 18   12 000AF         BNEQ    10$                                  ;
                 50              01   D0 000B1         MOVL    #1, I                                ; 0756
        51       01              50   78 000B4 8$:     ASHL    I, #1, R1                            ;
                 51              58   D1 000B8         CMPL    LENGTH, R1                           ;
                                 05   18 000BB         BGEQ    9$                                   ;
                 58              51   D0 000BD         MOVL    R1, LENGTH                           ; 0757
                                 07   11 000C0         BRB     10$                                  ;
        EE       50              09   F3 000C2 9$:     AOBLEQ  #9, I, 8$                            ; 0756
                 58              01   CE 000C6         MNEGL   #1, LENGTH                           ; 0755
        50  04   AC              03   C1 000C9 10$:    ADDL3   #3, DESCRIP, R0                      ; 0769
            BF   8F              60   91 000CE         CMPB    (R0), #191                           ;
                                 38   12 000D2         BNEQ    13$                                  ;
        50  04   AC              02   C1 000D4         ADDL3   #2, DESCRIP, R0                      ; 0772
                 15              60   91 000D9         CMPB    (R0), #21                            ;
                                 1D   12 000DC         BNEQ    11$                                  ;
            22   AE      0915    8F   B0 000DE         MOVW    #2325, TEMP_DSC+2                    ; 0777
            20   AE              52   B0 000E4         MOVW    R2, TEMP_DSC                         ; 0779
            24   AE        3C    AE   9E 000E8         MOVAB   TEMP_LEN, TEMP_DSC+4                 ; 0780
        50  04   AC              08   C1 000ED         ADDL3   #8, DESCRIP, R0                      ; 0781
            28   AE              60   90 000F2         MOVB    (R0), TEMP_DSC+8                     ;
                           20    AE   9F 000F6         PUSHAB  TEMP_DSC                             ; 0782
                                 03   11 000F9         BRB     12$                                  ;
                           3C    AE   9F 000FB 11$:    PUSHAB  TEMP_LEN                             ; 0785
                                 7E   D4 000FE 12$:    CLRL    -(SP)                                ;
                           04    AC   DD 00100         PUSHL   DESCRIP                              ;
           00000000G  00         03   FB 00103         CALLS   #3, BAS$$VA_FETCH                    ;
                                 7F   11 0010A         BRB     20$                                  ; 0771
        59  04   AC              02   C1 0010C 13$:    ADDL3   #2, DESCRIP, R9                      ; 0788
        16       06              69   8F 00111         CASEB   (R9), #6, #22                        ;
002E    003B     003B          003B      00115 14$:    .WORD   16$-14$,-                            ;
002E    002E     0046          003B      0011D                 16$-14$,-                            ;
002E    002E     002E          002E      00125                 16$-14$,-                            ;
0067    002E     002E          002E      0012D                 15$-14$,-                            ;
002E    0076     002E          002E      00135                 16$-14$,-                            ;
        0054     0046          002E      0013D                 17$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
                                                               15$-14$,-                            ;
```

```
                                                                19$-14$,-                                       ;
                                                                15$-14$,-                                       ;
                                                                15$-14$,-                                       ;
                                                                20$-14$,-                                       ;
                                                                15$-14$,-                                       ;
                                                                15$-14$,-                                       ;
                                                                17$-14$,-                                       ;
                                                                18$-14$                                         ;
                      7E        00G  8F  9A 00143 15$:   MOVZBL #BAS$K_DATTYPERR, -(SP)                         ; 0819
           00000000G  00             01  FB 00147        CALLS  #1, BAS$$STOP                                   ;
                                     3B  11 0014E        BRB    20$                                             ;
         50        04  AC            04  C1 00150 16$:   ADDL3  #4, DESCRIP, R0                                 ; 0791
                   3C  AE            90  D0 00155        MOVL   @(R0)+, TEMP_LEN                                ;
                                     30  11 00159        BRB    20$                                             ;
         51        04  AC            04  C1 0015B 17$:   ADDL3  #4, DESCRIP, R1                                 ; 0796
                       50            61  D0 00160        MOVL   (R1), R0                                        ;
                   3C  AE            60  7D 00163        MOVQ   (R0), TEMP_LEN                                  ;
                                     22  11 00167        BRB    20$                                             ; 0788
         51        04  AC            04  C1 00169 18$:   ADDL3  #4, DESCRIP, R1                                 ; 0804
                       50            61  D0 0016E        MOVL   (R1), R0                                        ;
                   3C  AE            60  7D 00171        MOVQ   (R0), TEMP_LEN                                  ;
                   44  AE       08   A0  7D 00175        MOVQ   8(R0), TEMP_LEN+8                               ; 0806
                                     0F  11 0017A        BRB    20$                                             ; 0788
         52                          02  C6 0017C 19$:   DIVL2  #2, R2                                          ; 0812
                                     52  D6 0017F        INCL   R2                                              ;
         5A        04  AC            04  C1 00181        ADDL3  #4, DESCRIP, R10                                ;
   3C    AE            9A            52  28 00186        MOVC3  R2, @(R10)+, TEMP_LEN                           ;
         54        04  AC            02  C1 0018B 20$:   ADDL3  #2, DESCRIP, R4                                 ; 0822
         16            06            64  8F 00190        CASEB  (R4), #6, #22                                   ;
00CB   003F          0038          0031     00194 21$:   .WORD  22$-21$,-                                       ;
00CB   00CB          004D          0046     0019C               23$-21$,-                                       ;
00CB   00CB          00CB          00CB     001A4               24$-21$,-                                       ;
0068   00CB          00CB          00CB     001AC               34$-21$,-                                       ;
00CB   007A          00CB          00CB     001B4               25$-21$,-                                       ;
       0060          0058          00CB     001BC               26$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                30$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                31$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                34$-21$,-                                       ;
                                                                27$-21$,-                                       ;
                                                                28$-21$                                         ;
                                   009A  31 001C2        BRW    34$                                             ; 0931
              1C  AE        3C       AE  98 001C5 22$:   CVTBL  TEMP_LEN, ARRAY_LEN                             ; 0825
                                     2E  11 001CA        BRB    29$                                             ;
              1C  AE        3C       AE  32 001CC 23$:   CVTWL  TEMP_LEN, ARRAY_LEN                             ; 0828
                                     27  11 001D1        BRB    29$                                             ;
```

```
                     1C     AE      3C   AE    D0 001D3 24$:     MOVL    TEMP_LEN, ARRAY_LEN                          ; 0831
                                          20    11 001D8          BRB     29$                                          ;
                     1C     AE      3C   AE    4A 001DA 25$:     CVTFL   TEMP_LEN, ARRAY_LEN                          ; 0834
                                          19    11 001DF          BRB     29$                                          ;
                     51     24      AE   9E    001E1 26$:        MOVAB   TEMP_DBL, R1                                 ; 0846
                     50     3C      AE   9E    001E5             MOVAB   TEMP_LEN, R0                                 ;
                                        00A4    31 001E9          BRW     42$                                          ;
                     1C     AE      3C  AE4AFD    001EC 27$:     CVTGL   TEMP_LEN, ARRAY_LEN                          ; 0854
                                          7D    11 001F2          BRB     36$                                          ;
                     1C     AE      3C  AE6AFD    001F4 28$:     CVTHL   TEMP_LEN, ARRAY_LEN                          ; 0857
                                          7C    11 001FA 29$:     BRB     38$                                          ;
             55     04      AC           08    C1 001FC 30$:     ADDL3   #8, DESCRIP, R5                              ; 0864
  00    3C   AE     04      BC           65    F8 00201          ASHP    (R5), @DESCRIP, TEMP_LEN, #0, #10, TEMP_P    ;
                    24      AE           0A       00208                                                                ;
                                        00B7    31 0020B          BRW     46$                                          ; 0865
             50     04      AC           03    C1 0020E 31$:     ADDL3   #3, DESCRIP, R0                              ; 0873
                    BF      8F           60    91 00213          CMPB    (R0), #191                                   ;
                                          0B    12 00217          BNEQ    32$                                          ;
                            7E      00G  8F    9A 00219          MOVZBL  #BAS$K_NOTIMP, -(SP)                         ; 0875
             00000000G      00           01    FB 0021D          CALLS   #1, BAS$$STOP                                ;
             50     04      AC           04    C1 00224 32$:     ADDL3   #4, DESCRIP, R0                              ; 0877
                            52           60    D0 00229          MOVL    (R0), ELEM_DESC                              ;
             16             06      02   A2    8F 0022C          CASEB   2(ELEM_DESC), #6, #22                        ; 0878
  002E      0049            0042        003B      00231 33$:     .WORD   35$-33$,-                                    ;
  002E      002E            0057        0050      00239                  37$-33$,-                                    ;
  002E      002E            002E        002E      00241                  39$-33$,-                                    ;
  008A      002E            002E        002E      00249                  34$-33$,-                                    ;
  002E      002E            002E        002E      00251                  40$-33$,-                                    ;
            0082            007A        002E      00259                  41$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         45$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         34$-33$,-                                    ;
                                                                         43$-33$,-                                    ;
                                                                         44$-33$                                      ;
                            7E      00G  8F    9A 0025F 34$:     MOVZBL  #BAS$K_DATTYPERR, -(SP)                      ; 0925
             00000000G      00           01    FB 00263          CALLS   #1, BAS$$STOP                                ;
                                          5F    11 0026A          BRB     47$                                          ;
                     1C     AE      04   B2    98 0026C 35$:     CVTBL   @4(ELEM_DESC), ARRAY_LEN                     ; 0882
                                          58    11 00271 36$:     BRB     47$                                          ; 0881
                     1C     AE      04   B2    32 00273 37$:     CVTWL   @4(ELEM_DESC), ARRAY_LEN                     ; 0886
                                          51    11 00278 38$:     BRB     47$                                          ; 0885
                     1C     AE      04   B2    D0 0027A 39$:     MOVL    @4(ELEM_DESC), ARRAY_LEN                     ; 0889
                                          4A    11 0027F          BRB     47$                                          ;
                     1C     AE      04   B2    4A 00281 40$:     CVTFL   @4(ELEM_DESC), ARRAY_LEN                     ; 0892
                                          43    11 00286          BRB     47$                                          ;
```

```
                          51        24    AE    9E 00288 41$:    MOVAB   TEMP_DBL, R1                                        ; 0901
                          50        04    A2    D0 0028C         MOVL    4(ELEM_DESC), R0                                    ;
                           00000000G  00    16 00290 42$:    JSB     BAS$$COPY_D_R1                                      ;
                          50        24    AE    7D 00296         MOVQ    TEMP_DBL, R0                                        ; 0902
                           00000000G  00    16 0029A         JSB     BAS$$SCALE_D_R1                                     ;
                    24    AE              50    7D 002A0         MOVQ    R0, TEMP_DBL                                        ; 0903
                    1C    AE        24    AE    6A 002A4         CVTDL   TEMP_DBL, ARRAY_LEN                                 ; 0905
                                          20    11 002A9         BRB     47$                                                 ; 0878
                    1C    AE        04    B24AFD 002AB 43$:    CVTGL   @4(ELEM_DESC), ARRAY_LEN                            ; 0909
                                          18    11 002B1         BRB     47$                                                 ;
                    1C    AE        04    B26AFD 002B3 44$:    CVTHL   @4(ELEM_DESC), ARRAY_LEN                            ; 0912
                                          10    11 002B9         BRB     47$                                                 ;
   00       04    B2              62        08    A2    F8 002BB 45$:    ASHP    8(ELEM_DESC), (ELEM_DESC), @4(ELEM_DESC), -  ; 0920
                    24    AE              0A       002C2                 #0, #10, TEMP_P                                     ;
            1C    AE      24    AE              0A    36 002C5 46$:    CVTPL   #10, TEMP_P, ARRAY_LEN                              ; 0921
                    58    AE  020E0000    8F    D0 002CB 47$:    MOVL    #34471936, TEMP_STR_DESC                            ; 0940
                                    5C    AE    D4 002D3         CLRL    TEMP_STR_DESC+4                                     ; 0941
                                    58    AE    9F 002D6         PUSHAB  TEMP_STR_DESC                                       ; 0942
                                    20    AE    9F 002D9         PUSHAB  ARRAY_LEN                                           ;
              00000000G  00               02    FB 002DC         CALLS   #2, STR$GET1_DX                                     ;
                         09               50    E8 002E3         BLBS    STR_STATUS, 48$                                     ; 0943
                                          50    DD 002E6         PUSHL   STR_STATUS                                          ; 0945
              00000000G  00               01    FB 002E8         CALLS   #1, BAS$$STOP                                       ;
                         54         5C    AE    D0 002EF 48$:    MOVL    TEMP_STR_DESC+4, STR_BUF                            ; 0946
            18    AE     57         0C    AE    C3 002F3         SUBL3   INDEX_INCR, LOW_INDEX, 24(SP)                       ; 0957
                    14    AE        0C BE46     9E 002F9         MOVAB   @INDEX_INCR[HIGH_INDEX], 20(SP)                     ; 0961
                                          57    D4 002FF         CLRL    INDEX                                               ; 0974
                                        0284    31 00301         BRW     83$                                                 ;
                          55        18    AE    D0 00304 49$:    MOVL    24(SP), INDEX_NUMBER                                ; 0957
                    04    AE              57    D0 00308         MOVL    INDEX, INDEX_VALUE                                  ; 0958
                                          56    D4 0030C         CLRL    VALUE_LOCATION                                      ; 0959
                          55        0C    AE    C0 0030E 50$:    ADDL2   INDEX_INCR, INDEX_NUMBER                            ; 0961
                    14    AE              55    D1 00312         CMPL    INDEX_NUMBER, 20(SP)                                ;
                                          45    13 00316         BEQL    53$                                                 ;
                50        55              01    78 00318         ASHL    #1, INDEX_NUMBER, R0                                ; 0963
                51  08    AE              08    C3 0031C         SUBL3   #8, BOUNDS, R1                                      ;
                        6140        04    AE    D1 00321         CMPL    INDEX_VALUE, (R1)[R0]                               ;
                                          0C    19 00326         BLSS    51$                                                 ;
                51  08    AE              04    C3 00328         SUBL3   #4, BOUNDS, R1                                      ; 0964
                        6140        04    AE    D1 0032D         CMPL    INDEX_VALUE, (R1)[R0]                               ;
                                          15    15 00332         BLEQ    52$                                                 ;
                                    58    AE    9F 00334 51$:    PUSHAB  TEMP_STR_DESC                                       ; 0967
              00000000G  00               01    FB 00337         CALLS   #1, STR$FREE1_DX                                    ;
                         7E        00G    8F    9A 0033E         MOVZBL  #BAS$K_SUBOUTRAN, -(SP)                             ; 0968
              00000000G  00               01    FB 00342         CALLS   #1, BAS$$STOP                                       ;
                51  10    AE              04    C3 00349 52$:    SUBL3   #4, MULTIPLIERS, R1                                 ; 0970
                50        56            6145    C5 0034E         MULL3   (R1)[INDEX_NUMBER], VALUE_LOCATION, R0              ;
                56        50        04    AE    C1 00353         ADDL3   INDEX_VALUE, R0, VALUE_LOCATION                     ;
                                    04    AE    D4 00358         CLRL    INDEX_VALUE                                         ; 0971
                                          B1    11 0035B         BRB     50$                                                 ; 0961
                50        56              58    C5 0035D 53$:    MULL3   LENGTH, VALUE_LOCATION, R0                          ; 0974
                51  04    AC              10    C1 00361         ADDL3   #16, DESCRIP, R1                                    ;
                56        50              61    C1 00366         ADDL3   (R1), R0, VALUE_LOCATION                            ;
                                          51    D4 0036A         CLRL    R1                                                  ; 0982
                50  04    AC              02    C1 0036C         ADDL3   #2, DESCRIP, R0                                     ;
                          18              60    91 00371         CMPB    (R0), #24                                           ;
                                          30    12 00374         BNEQ    56$                                                 ;
```

```
                                    51  D6 00376       INCL    R1                                          :
               4C   AE              66  B0 00378       MOVW    (VALUE_LOCATION), VALUE_DESCR               : 0989
               4E   AE       02     A6  90 0037C       MOVB    2(VALUE_LOCATION), VALUE_DESCR+2            : 0990
                    02       03     A6  91 00381       CMPB    3(VALUE_LOCATION), #2                       : 0991
                                    05  12 00385       BNEQ    54$                                         :
                    50              01  D0 00387       MOVL    #1, R0                                      :
                                    04  11 0038A       BRB     55$                                         :
                    50       03     A6  9A 0038C 54$:  MOVZBL  3(VALUE_LOCATION), R0                       : 0992
               4F   AE              50  90 00390 55$:  MOVB    R0, VALUE_DESCR+3                           : 0991
               50   AE       04     A6  D0 00394       MOVL    4(VALUE_LOCATION), VALUE_DESCR+4            : 0993
                    15       4E     AE  91 00399       CMPB    VALUE_DESCR+2, #21                          : 0994
                                    2C  12 0039D       BNEQ    57$                                         :
               54   AE       08     A6  90 0039F       MOVB    8(VALUE_LOCATION), VALUE_DESCR+8            : 0999
                                    25  11 003A4       BRB     57$                                         : 0982
               4C   AE       04     BC  B0 003A6 56$:  MOVW    @DESCRIP, VALUE_DESCR                       : 1004
       50      04   AC              02  C1 003AB       ADDL3   #2, DESCRIP, R0                             : 1005
               4E   AE              60  90 003B0       MOVB    (R0), VALUE_DESCR+2                         :
               4F   AE              01  90 003B4       MOVB    #1, VALUE_DESCR+3                           : 1006
               50   AE              56  D0 003B8       MOVL    VALUE_LOCATION, VALUE_DESCR+4               : 1007
                    15       4E     AE  91 003BC       CMPB    VALUE_DESCR+2, #21                          : 1008
                                    09  12 003C0       BNEQ    57$                                         :
       50      04   AC              08  C1 003C2       ADDL3   #8, DESCRIP, R0                             : 1013
               54   AE              60  90 003C7       MOVB    (R0), VALUE_DESCR+8                         :
       50      04   AC              03  C1 003CB 57$:  ADDL3   #3, DESCRIP, R0                             : 1021
               BF   8F              60  91 003D0       CMPB    (R0), #191                                  :
                                    56  12 003D4       BNEQ    61$                                         :
                    15              51  E9 003D6       BLBC    R1, 58$                                     : 1025
                             58     AE  9F 003D9       PUBHAB  TEMP_STR_DESC                               : 1028
        00000000G   00              01  FB 003DC       CALLS   #1, STR$FREE1_DX                            :
                    7E      00G     8F  9A 003E3       MOVZBL  #BAS$K_NOTIMP, -(SP)                        : 1029
        00000000G   00              01  FB 003E7       CALLS   #1, BAS$$STOP                               :
       50      04   AC              02  C1 003EE 58$:  ADDL3   #2, DESCRIP, R0                             : 1032
                    15              60  91 003F3       CMPB    (R0), #21                                   :
                                    1E  12 003F6       BNEQ    59$                                         :
               22   AE     0915     8F  B0 003F8       MOVW    #2325, TEMP_DSC+2                           : 1037
               20   AE       04     BC  B0 003FE       MOVW    @DESCRIP, TEMP_DSC                          : 1039
               24   AE       2C     AE  9E 00403       MOVAB   TEMP_BUF, TEMP_DSC+4                        : 1040
       50      04   AC              08  C1 00408       ADDL3   #8, DESCRIP, R0                             : 1041
               28   AE              60  90 0040D       MOVB    (R0), TEMP_DSC+8                            :
                             20     AE  9F 00411       PUSHAB  TEMP_DSC                                    : 1042
                                    03  11 00414       BRB     60$                                         :
                             2C     AE  9F 00416 59$:  PUSHAB  TEMP_BUF                                    : 1045
                                    56  DD 00419 60$:  PUSHL   VALUE_LOCATION                              :
                             04     AC  DD 0041B       PUSHL   DESCRIP                                     :
        00000000G   00              03  FB 0041E       CALLS   #3, BAS$$VA_FETCH                           :
               50   AE       2C     AE  9E 00425       MOVAB   TEMP_BUF, VALUE_DESCR+4                     : 1047
                                    1F  11 0042A       BRB     62$                                         : 1021
       50      04   AC              03  C1 0042C 61$:  ADDL3   #3, DESCRIP, R0                             : 1052
                    04              60  91 00431       CMPB    (R0), #4                                    :
                                    15  13 00434       BEQL    62$                                         :
                             58     AE  9F 00436       PUSHAB  TEMP_STR_DESC                               : 1055
        00000000G   00              01  FB 00439       CALLS   #1, STR$FREE1_DX                            :
                    7E      00G     8F  9A 00440       MOVZBL  #BAS$K_NOTIMP, -(SP)                        : 1056
        00000000G   00              01  FB 00444       CALLS   #1, BAS$$STOP                               :
       16           06       4E     AE  8F 0044B 62$:  CASEB   VALUE_DESCR+2, #6, #22                      : 1063
 00C4  0031            0031       0031     00450 63$:  .WORD   64$-63$,-                                   :
 00C4  00C4            0040       003A     00458               64$-63$,-                                   :
```

```
    00C4      00C4      00C4      00C4  00460              64$-63$,-                  :
    0123      00C4      00C4      00C4  00468              73$-63$,-                  :
    00C4      0071      00C4      00C4  00470              65$-63$,-                  :
              0069      0062      00C4  00478              66$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           81$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           70$-63$,-                  :
                                                           73$-63$,-                  :
                                                           73$-63$,-                  :
                                                           67$-63$,-                  :
                                                           68$-63$                    :
                          0093   31 0047E        BRW      73$                         : 1184
          FF A744     50    BE   90 00481 64$:   MOVB     @VALUE_DESCR+4, -1(INDEX)[STR_BUF]  : 1067
                          00FE   31 00487        BRW      83$                         :
               6E     50    BE   4A 0048A 65$:   CVTFL    @VALUE_DESCR+4, STR_BUF_LONG  : 1071
                            2E   11 0048E        BRB      69$                         : 1072
               51     24    AE   9E 00490 66$:   MOVAB    TEMP_DBL, R1                : 1085
               50     50    AE   D0 00494        MOVL     VALUE_DESCR+4, R0           :
                  00000000G 00   16 00498        JSB      BAS$$COPY_D_R1              :
               50     24    AE   7D 0049E        MOVQ     TEMP_DBL, R0                : 1086
                  00000000G 00   16 004A2        JSB      BAS$DSCALE_D_R1             :
          24   AE           50   7D 004A8        MOVQ     R0, TEMP_DBL                : 1087
               6E     24    AE   6A 004AC        CVTDL    TEMP_DBL, STR_BUF_LONG      : 1089
                            0C   11 004B0        BRB      69$                         : 1090
               6E     50  BE4AFD    004B2 67$:   CVTGL    @VALUE_DESCR+4, STR_BUF_LONG  : 1095
                            05   11 004B7        BRB      69$                         : 1096
               6E     50  BE6AFD    004B9 68$:   CVTHL    @VALUE_DESCR+4, STR_BUF_LONG  : 1101
                          00C2   31 004BE 69$:   BRW      82$                         : 1102
      50      04   AC           03   C1 004C1 70$:   ADDL3    #3, DESCRIP, R0         : 1119
              BF   8F           60   91 004C6        CMPB     (R0), #191              :
                                15   12 004CA        BNEQ     71$                     :
                      58    AE   9F 004CC        PUSHAB   TEMP_STR_DESC               : 1122
          00000000G   00        01   FB 004CF        CALLS    #1, STR$FREE1_DX        :
                      7E   00G  8F   9A 004D6        MOVZBL   #BAS$K_NOTIMP, -(SP)    : 1123
          00000000G   00        01   FB 004DA        CALLS    #1, BAS$$STUP           :
              16      06   4E   AE   8F 004E1 71$:   CASEB    VALUE_DESCR+2, #6, #22  : 1126
    002E      0051      004B      0045  004E6 72$:   .WORD    74$-72$,-               :
    002E      002E      005D      0057  004EE              75$-72$,-                  :
    002E      002E      002E      002E  004F6              76$-72$,-                  :
    008D      002E      002E      002E  004FE              73$-72$,-                  :
    002E      002E      002E      002E  00506              77$-72$,-                  :
              0086      007F      002E  0050E              78$-72$,-                  :
                                                           73$-72$,-                  :
                                                           73$-72$,-                  :
                                                           73$-72$,-                  :
                                                           73$-72$,-                  :
                                                           73$-72$,-                  :
```

```
                                                        73$-72$,-
                                                        73$-72$,-
                                                        73$-72$,-
                                                        73$-72$,-
                                                        81$-72$,-
                                                        73$-72$,-
                                                        73$-72$,-
                                                        73$-72$,-
                                                        73$-72$,-
                                                        73$-72$,-
                                                        79$-72$,-
                                                        80$-72$
                    58   AE   9F 00514 73$:    PUSHAB  TEMP_STR_DESC                                   ;  1175
00000000G   00          01   FB 00517          CALLS   #1, STR$FREE1_DX
                7E      00G  8F   9A 0051E     MOVZBL  #BAS$K_DATTYPERR, -(SP)                         ;  1176
00000000G   00          01   FB 00522          CALLS   #1, BAS$$STOP
                        5D   11 00529          BRB     83$                                             ;  1126
                6E      50   BE   98 0052B 74$:  CVTBL  @VALUE_DESCR+4, STR_BUF_LONG                   ;  1130
                        57   11 0052F          BRB     83$                                             ;  1129
                6E      50   BE   32 00531 75$:  CVTWL  @VALUE_DESCR+4, STR_BUF_LONG                   ;  1134
                        51   11 00535          BRB     83$                                             ;  1133
                6E      50   BE   D0 00537 76$:  MOVL   @VALUE_DESCR+4, STR_BUF_LONG                   ;  1137
                        4B   11 0053B          BRB     83$
                6E      50   BE   4A 0053D 77$:  CVTFL  @VALUE_DESCR+4, STR_BUF_LONG                   ;  1140
                        45   11 00541          BRB     83$
                51      24   AE   9E 00543 78$:  MOVAB  TEMP_DBL, R1                                   ;  1149
                50      50   AE   D0 00547          MOVL   VALUE_DESCR+4, R0
                    00000000G   00   16 0054B      JSB    BAS$$COPY_D_R1
                50      24   AE   7D 00551          MOVQ   TEMP_DBL, R0                              ;  1150
                    00000000G   00   16 00555      JSB    BAS$$SCALE_D_R1
            24  AE          50   7D 0055B          MOVQ   R0, TEMP_DBL                              ;  1151
                6E      24   AE   6A 0055F          CVTDL  TEMP_DBL, STR_BUF_LONG                    ;  1153
                        23   11 00563          BRB     83$                                             ;  1126
                6E      50   BE4AFD 00565 79$:  CVTGL  @VALUE_DESCR+4, STR_BUF_LONG                   ;  1157
                        1C   11 0056A          BRB     83$
                6E      50   BE6AFD 0056C 80$:  CVTHL  @VALUE_DESCR+4, STR_BUF_LONG                   ;  1160
                        15   11 00571          BRB     83$
        00      50  BE      4C   AE      54   AE   F8 00573 81$:  ASHP   VALUE_DESCR+8, VALUE_DESCR, -  ;  1166
                            24   AE           0A        0057B          @VALUE_DESCR+4, #0, #10, TEMP_P
                6E          24   AE           0A   3C   0057E          CVTPL  #10, TEMP_P, STR_BUF_LONG ;  1169
                            FF A744           6E   90   00583 82$:     MOVB   STR_BUF_LONG, -1[INDEX][STR_BUF]  ;  1170
        FD75        57          01       1C   AE   F1   00588 83$:     ACBL   ARRAY_LEN, #1, INDEX, 49$ ;  0955
                                             58   AE   9F   0058F          PUSHAB TEMP_STR_DESC            ;  1195
                                             08   AC   DD   00592          PUSHL  STR_DESC
                    00000000G   00           02   FB   00595          CALLS  #2, STR$COPY_DX
                                             58   AE   9F   0059C          PUSHAB TEMP_STR_DESC            ;  1199
                    00000000G   00           01   FB   0059F          CALLS  #1, STR$FREE1_DX
                                             04   005A6               RET                              ;  1201

; Routine Size:  1447 bytes,    Routine Base:  _BAS$CODE + 0018

;  834          1202  1
```

```
;  836     1203  1  ROUTINE STORE (                                  ! Store string elements into array
;  837     1204  1          STR_DESC,                                 ! Where to find the string
;  838     1205  1          DESCRIP                                   ! The array to store it in
;  839     1206  1      ) : NOVALUE =
;  840     1207  1
;  841     1208  1  !++
;  842     1209  1  ! FUNCTIONAL DESCRIPTION:
;  843     1210  1  !
;  844     1211  1  !       Store string elements into an array.  The array will be numeric.
;  845     1212  1  !
;  846     1213  1  ! FORMAL PARAMETERS:
;  847     1214  1  !
;  848     1215  1  !       STR_DESC.rx.dx  The place from which to get the string values
;  849     1216  1  !       DESCRIP.rx.da   The descriptor of the array or virtual array
;  850     1217  1  !
;  851     1218  1  ! IMPLICIT INPUTS:
;  852     1219  1  !
;  853     1220  1  !       NONE
;  854     1221  1  !
;  855     1222  1  ! IMPLICIT OUTPUTS:
;  856     1223  1  !
;  857     1224  1  !       NONE
;  858     1225  1  !
;  859     1226  1  ! ROUTINE VALUE:
;  860     1227  1  ! COMPLETION CODES:
;  861     1228  1  !
;  862     1229  1  !       NONE
;  863     1230  1  !
;  864     1231  1  ! SIDE EFFECTS:
;  865     1232  1  !
;  866     1233  1  !       Signals if an error is encountered.
;  867     1234  1  !
;  868     1235  1  !--
;  869     1236  1
;  870     1237  2      BEGIN
;  871     1238  2
;  872     1239  2      GLOBAL REGISTER
;  873     1240  2          BSF$A_MAJOR_STG = 11,
;  874     1241  2          BSF$A_MINOR_STG = 10,
;  875     1242  2          BSF$A_TEMP_STG = 9;
;  876     1243  2
;  877     1244  2      BUILTIN
;  878     1245  2          ASHP,
;  879     1246  2          CVTLF,
;  880     1247  2          CVTLD,
;  881     1248  2          CVTLG,
;  882     1249  2          CVTLM,
;  883     1250  2          CVTLP;
;  884     1251  2
;  885     1252  2      LOCAL
;  886     1253  2          INDEX_VALUE,
;  887     1254  2          VALUE_LOCATION,
;  888     1255  2          MULTIPLIERS : REF VECTOR,
;  889     1256  2          BOUNDS : REF VECTOR,
;  890     1257  2          LOW_INDEX,
;  891     1258  2          HIGH_INDEX,
;  892     1259  2          INDEX_INCR,
```

```
;  893          1260  2          INDEX_NUMBER,
;  894          1261  2          INDEX_ERROR : INITIAL (0),
;  895          1262  2          VALUE_DESCR : BLOCK [12, BYTE],
;  896          1263  2          LENGTH,
;  897          1264  2          STR_BUF : REF VECTOR [255,BYTE],
;  898          1265  2          STR_BUF_LONG,
;  899          1266  2          TEMP_BUF : VECTOR [4];
;  900          1267  2
;  901          1268  2      LABEL
;  902          1269  2          INCR_LOOP;
;  903          1270  2
;  904          1271  2      MAP
;  905          1272  2          DESCRIP : REF BLOCK [8, BYTE],
;  906          1273  2          STR_DESC : REF BLOCK [8, BYTE];
;  907          1274  2
;  908          1275  2      STR_BUF = .STR_DESC [DSC$A_POINTER];
;  909          1276  2
;  910          1277  2  !+
;  911          1278  2  ! The coefficients and bounds must be present.
;  912          1279  2  !-
;  913          1280  2
;  914          1281  2      IF ( NOT (.DESCRIP [DSC$V_FL_COEFF] AND .DESCRIP [DSC$V_FL_BOUNDS])) THEN BAS$$STOP (BAS$K_ARGDONMAT);
;  915          1282  2
;  916          1283  2      MULTIPLIERS = DESCRIP [DSC$L_M1];
;  917          1284  2      BOUNDS = DESCRIP [DSC$L_M1] + (%UPVAL*.DESCRIP [DSC$B_DIMCT]);
;  918          1285  2  !+
;  919          1286  2  ! Compute the lower and upper index numbers based on how the array
;  920          1287  2  ! is stored.
;  921          1288  2  !-
;  922          1289  2
;  923          1290  2      IF (.DESCRIP [DSC$V_FL_COLUMN])
;  924          1291  2      THEN
;  925          1292  2          BEGIN
;  926          1293  2          LOW_INDEX = .DESCRIP [DSC$B_DIMCT];
;  927          1294  2          HIGH_INDEX = 1;
;  928          1295  2          INDEX_INCR = -1;
;  929          1296  2          END
;  930          1297  2      ELSE
;  931          1298  2          BEGIN
;  932          1299  2          LOW_INDEX = 1;
;  933          1300  2          HIGH_INDEX = .DESCRIP [DSC$B_DIMCT];
;  934          1301  2          INDEX_INCR = 1;
;  935          1302  2          END;
;  936          1303  2
;  937          1304  2  !+
;  938          1305  2  ! If this is a decimal array, the length in the descriptor is the number of
;  939          1306  2  ! 4 bit digits (not including the sign).  Convert this length to the number
;  940          1307  2  ! of bytes.
;  941          1308  2  ! Also, if this is a virtual array, the size must be a multiple of 2.  This
;  942          1309  2  ! is true for arrays of records as well.
;  943          1310  2  !-
;  944          1311  2      CASE .DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  945          1312  2          SET
;  946          1313  2
;  947          1314  2          [DSC$K_DTYPE_P] :                   ! decimal
;  948          1315  3              BEGIN
;  949          1316  3              LENGTH = (.DESCRIP [DSC$W_LENGTH]/2) + 1;
```

```
;  950          1317  3            IF .DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA
;  951          1318  3            THEN
;  952          1319  4                BEGIN
;  953          1320  4
;  954          1321  5                LENGTH = ( INCR I FROM 1 TO 9 BY 1 DO
;  955          1322  6                               IF .LENGTH LSS (1 ^ .I)
;  956          1323  6                               THEN EXITLOOP (1 ^ .I) );
;  957          1324  5                END;
;  958          1325  4            END;
;  959          1326  2
;  960          1327  2        [INRANGE, OUTRANGE] :
;  961          1328  2            LENGTH = .DESCRIP [DSC$W_LENGTH];
;  962          1329  2        TES;
;  963          1330  2
;  964          1331  2    !+
;  965          1332  2    ! Calculate the linear index.  CHANGE operates only on row 0, so all indicies
;  966          1333  2    ! except one will be zero.  This code should accomodate FORTRAN arrays.
;  967          1334  2    !-
;  968          1335  2
;  969          1336  2        INCR INDEX FROM 1 TO .STR_DESC [DSC$W_LENGTH] DO
;  970          1337  2        INCR_LOOP:
;  971          1338  2        BEGIN
;  972          1339  3                STR_BUF_LONG = .STR_BUF [.INDEX - 1];
;  973          1340  3                INDEX_NUMBER = .LOW_INDEX - .INDEX_INCR;
;  974          1341  3                INDEX_VALUE = .INDEX;
;  975          1342  3                VALUE_LOCATION = 0;
;  976          1343  3
;  977          1344  3                WHILE ((INDEX_NUMBER = .INDEX_NUMBER + .INDEX_INCR) NEQ (.HIGH_INDEX + .INDEX_INCR)) DO
;  978          1345  4                BEGIN
;  979          1346  6                IF ((.INDEX_VALUE LSS .BOUNDS [(.INDEX_NUMBER - 1)*2])      !
;  980          1347  5                    OR (.INDEX_VALUE GTR .BOUNDS [((.INDEX_NUMBER - 1)*2) + 1]))
;  981          1348  4                THEN
;  982          1349  5                    BEGIN
;  983          1350  5
;  984          1351  5                    INDEX_ERROR = .INDEX;
;  985          1352  5                    LEAVE INCR_LOOP;
;  986          1353  5
;  987          1354  4                    END;
;  988          1355  4
;  989          1356  4                VALUE_LOCATION = (.VALUE_LOCATION*.MULTIPLIERS [.INDEX_NUMBER - 1]) + .INDEX_VALUE;
;  990          1357  4                INDEX_VALUE = 0;                                ! all subsequent indicies zero
;  991          1358  3                END;
;  992          1359  3
;  993          1360  3            VALUE_LOCATION = (.VALUE_LOCATION*.LENGTH) + .DESCRIP [DSC$A_A0];
;  994          1361  3    !+
;  995          1362  3    ! Build a descriptor pointing to the value cell in the array.  If this
;  996          1363  3    ! is an array of descriptors, the descriptor is copied, otherwise it
;  997          1364  3    ! is constructed.
;  998          1365  3    !-
;  999          1366  3
; 1000          1367  4        IF (.DESCRIP [DSC$B_DTYPE] EQLU DSC$K_DTYPE_DSC)
; 1001          1368  3        THEN
; 1002          1369  4            BEGIN
; 1003          1370  4
; 1004          1371  4            MAP
; 1005          1372  4                VALUE_LOCATION : REF BLOCK [8, BYTE];
; 1006          1373  4
```

```
;  1007          1374  4          VALUE_DESCR [DSC$W_LENGTH] = .VALUE_LOCATION [DSC$W_LENGTH];
;  1008          1375  4          VALUE_DESCR [DSC$B_DTYPE] = .VALUE_LOCATION [DSC$B_DTYPE];
;  1009          1376  5          VALUE_DESCR [DSC$B_CLASS] = (IF (.VALUE_LOCATION [DSC$B_CLASS] EQLU DSC$K_CLASS_D) THEN DSC$K_CLASS_
;  1010          1377  4              ELSE .VALUE_LOCATION [DSC$B_CLASS]);
;  1011          1378  4          VALUE_DESCR [DSC$A_POINTER] = .VALUE_LOCATION [DSC$A_POINTER];
;  1012          1379  4          IF .VALUE_DESCR [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  1013          1380  4          THEN
;  1014          1381  5              BEGIN
;  1015          1382  5              MAP
;  1016          1383  5                  VALUE_LOCATION : REF BLOCK [12,BYTE];
;  1017          1384  5              VALUE_DESCR [DSC$B_SCALE] = .VALUE_LOCATION [DSC$B_SCALE];
;  1018          1385  4              END;
;  1019          1386  4      END
;  1020          1387  3      ELSE
;  1021          1388  4          BEGIN
;  1022          1389  4          VALUE_DESCR [DSC$W_LENGTH] = .DESCRIP [DSC$W_LENGTH];
;  1023          1390  4          VALUE_DESCR [DSC$B_DTYPE] = .DESCRIP [DSC$B_DTYPE];
;  1024          1391  4          VALUE_DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
;  1025          1392  4          VALUE_DESCR [DSC$A_POINTER] = .VALUE_LOCATION;
;  1026          1393  5          IF (.VALUE_DESCR [DSC$B_DTYPE] EQL DSC$K_DTYPE_P)
;  1027          1394  4          THEN
;  1028          1395  5              BEGIN
;  1029          1396  5              MAP
;  1030          1397  5                  DESCRIP : REF BLOCK [12,BYTE];
;  1031          1398  5              VALUE_DESCR [DSC$B_SCALE] = .DESCRIP [DSC$B_SCALE];
;  1032          1399  4              END;
;  1033          1400  3          END;
;  1034          1401  3
;  1035          1402  4      IF (.DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA)
;  1036          1403  4      THEN
;  1037          1404  4          VALUE_DESCR [DSC$A_POINTER] = TEMP_BUF [0];
;  1038          1405  4
;  1039          1406  4  !+
;  1040          1407  4  ! Copy the string element to the array.  The longword element must stored as
;  1041          1408  4  ! the data type of the array.  (Note that longword is used because the
;  1042          1409  4  ! instructions are BUILTINs.)
;  1043          1410  4  !-
;  1044          1411  4
;  1045          1412  4          CASE .DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  1046          1413  4              SET
;  1047          1414  4
;  1048          1415  4              [DSC$K_DTYPE_B] :
;  1049          1416  4                  BLOCK [.VALUE_DESCR [DSC$A_POINTER], 0, 0, %BPUNIT, 1]
;  1050          1417  4                  = .STR_BUF_LONG;
;  1051          1418  4
;  1052          1419  4              [DSC$K_DTYPE_W] :
;  1053          1420  4                  BLOCK [.VALUE_DESCR [DSC$A_POINTER], 0, 0, %BPVAL/2, 1]
;  1054          1421  4                  = .STR_BUF_LONG;
;  1055          1422  4
;  1056          1423  4              [DSC$K_DTYPE_L] :
;  1057          1424  4                  .VALUE_DESCR [DSC$A_POINTER] = .STR_BUF_LONG;
;  1058          1425  4
;  1059          1426  4              [DSC$K_DTYPE_F] :                   ! 32-bit floating point
;  1060          1427  4                  CVTLF (STR_BUF_LONG, .VALUE_DESCR [DSC$A_POINTER]);
;  1061          1428  4
;  1062          1429  4              [DSC$K_DTYPE_D] :                   ! 64-bit double floating
;  1063          1430  4                  BEGIN
```

```
;  1064          1431  4                !+
;  1065          1432  4                ! Apply scale to double value.
;  1066          1433  4                !-
;  1067          1434  4                LOCAL
;  1068          1435  4                    TEMP_DBL : VECTOR [2];
;  1069          1436  4                REGISTER
;  1070          1437  4                    R0 = 0,
;  1071          1438  4                    R1 = 1;
;  1072          1439  4                CVTLD (STR_BUF_LONG, TEMP_DBL);
;  1073          1440  4                BAS$SCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  1074          1441  4                TEMP_DBL [0] = .R0;
;  1075          1442  4                TEMP_DBL [1] = .R1;
;  1076          1443  4                BAS$$COPY_D_R1 (TEMP_DBL [0], .VALUE_DESCR [DSC$A_POINTER]);
;  1077          1444  3                END;
;  1078          1445  3
;  1079          1446  3            [DSC$K_DTYPE_G] :                   ! G floating
;  1080          1447  3                CVTLG (STR_BUF_LONG, .VALUE_DESCR [DSC$A_POINTER]);
;  1081          1448  3
;  1082          1449  3            [DSC$K_DTYPE_H] :                   ! H floating
;  1083          1450  3                CVTLH (STR_BUF_LONG, .VALUE_DESCR [DSC$A_POINTER]);
;  1084          1451  3
;  1085          1452  3            [DSC$K_DTYPE_P] :                   ! decimal
;  1086          1453  4                BEGIN
;  1087          1454  4                LOCAL
;  1088          1455  4                    TEMP_P : VECTOR [6, BYTE];
;  1089          1456  4
;  1090          1457  4                CVTLP (STR_BUF_LONG, %REF(10), TEMP_P);
;  1091          1458  4                ASHP (%REF(-.VALUE_DESCR [DSC$B_SCALE]), %REF(10),
;  1092          1459  4                    TEMP_P, %REF(0), VALUE_DESCR [DSC$W_LENGTH],
;  1093          1460  4                    .VALUE_DESCR [DSC$A_POINTER]);
;  1094          1461  3                END;
;  1095          1462  3
;  1096          1463  3            [DSC$K_DTYPE_DSC] :
;  1097          1464  4                BEGIN
;  1098          1465  4
;  1099          1466  4                IF .DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA
;  1100          1467  4                THEN
;  1101          1468  4                    BAS$$STOP (BAS$K_NOTIMP);   ! no virtual dyn mapped arrays
;  1102          1469  4
;  1103          1470  4                CASE .VALUE_DESCR [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  1104          1471  4                SET
;  1105          1472  4                    [DSC$K_DTYPE_B] :
;  1106          1473  4                        BLOCK [.VALUE_DESCR [DSC$A_POINTER], 0, 0, %BPUNIT, 1]
;  1107          1474  4                        = .STR_BUF_LONG;
;  1108          1475  4
;  1109          1476  4                    [DSC$K_DTYPE_W] :
;  1110          1477  4                        BLOCK [.VALUE_DESCR [DSC$A_POINTER], 0, 0, %BPVAL/2, 1]
;  1111          1478  4                        = .STR_BUF_LONG;
;  1112          1479  4
;  1113          1480  4                    [DSC$K_DTYPE_L] :
;  1114          1481  4                        .VALUE_DESCR [DSC$A_POINTER] = .STR_BUF_LONG;
;  1115          1482  4
;  1116          1483  4                    [DSC$K_DTYPE_F] :                   ! 32-bit floating point
;  1117          1484  4                        CVTLF (STR_BUF_LONG, .VALUE_DESCR [DSC$A_POINTER]);
;  1118          1485  4
;  1119          1486  4                    [DSC$K_DTYPE_D] :                   ! 64-bit double floating
;  1120          1487  5                        BEGIN
```

```
;  1121          1488  5                    !+
;  1122          1489  5                    ! Apply scale to double value.
;  1123          1490  5                    !-
;  1124          1491  5                    LOCAL
;  1125          1492  5                        TEMP_DBL : VECTOR [2];
;  1126          1493  5                    REGISTER
;  1127          1494  5                        R0 = 0,
;  1128          1495  5                        R1 = 1;
;  1129          1496  5                    CVTLD (STR_BUF_LONG, TEMP_DBL);
;  1130          1497  5                    BAS$SCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  1131          1498  5                    TEMP_DBL [0] = .R0;
;  1132          1499  5                    TEMP_DBL [1] = .R1;
;  1133          1500  5                    BAS$COPY_D_R1 (TEMP_DBL [0], .VALUE_DESCR [DSC$A_POINTER]);
;  1134          1501  4                    END;
;  1135          1502  4
;  1136          1503  4                [DSC$K_DTYPE_G] :                   ! G floating
;  1137          1504  4                    CVTLG (STR_BUF_LONG, .VALUE_DESCR [DSC$A_POINTER]);
;  1138          1505  4
;  1139          1506  4                [DSC$K_DTYPE_H] :                   ! H floating
;  1140          1507  4                    CVTLH (STR_BUF_LONG, .VALUE_DESCR [DSC$A_POINTER]);
;  1141          1508  4
;  1142          1509  4                [DSC$K_DTYPE_P] :                   ! decimal
;  1143          1510  5                    BEGIN
;  1144          1511  5                    LOCAL
;  1145          1512  5                        TEMP_P : VECTOR [6, BYTE];
;  1146          1513  5
;  1147          1514  5                    CVTLP (STR_BUF_LONG, %REF(10), TEMP_P);
;  1148          1515  5                    ASHP (%REF(-.VALUE_DESCR [DSC$B_SCALE]), %REF(10),
;  1149          1516  5                        TEMP_P, %REF(0), VALUE_DESCR [DSC$W_LENGTH],
;  1150          1517  5                        .VALUE_DESCR [DSC$A_POINTER]);
;  1151          1518  4                    END;
;  1152          1519  4
;  1153          1520  4                [INRANGE, OUTRANGE] :
;  1154          1521  4                    BAS$$STOP (BAS$K_DATTYPERR);
;  1155          1522  4
;  1156          1523  4                TES;
;  1157          1524  3                END;
;  1158          1525  3
;  1159          1526  3            [INRANGE, OUTRANGE] :
;  1160          1527  3                BAS$$STOP (BAS$K_DATTYPERR);
;  1161          1528  3
;  1162          1529  3            TES;
;  1163          1530  3
;  1164          1531  4        IF (.DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA)
;  1165          1532  3        THEN
;  1166          1533  4            BEGIN
;  1167          1534  4
;  1168          1535  4            IF (.DESCRIP [DSC$B_DTYPE] EQLU DSC$K_DTYPE_DSC) THEN BAS$$STOP (BAS$K_NOTIMP);
;  1169          1536  4
;  1170          1537  4            IF .DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  1171          1538  4            THEN
;  1172          1539  5                BEGIN
;  1173          1540  5                LOCAL
;  1174          1541  5                    TEMP_DSC : BLOCK [12, BYTE];
;  1175          1542  5                TEMP_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_P;
;  1176          1543  5                TEMP_DSC [DSC$B_CLASS] = DSC$K_CLASS_SD;
;  1177          1544  5                TEMP_DSC [DSC$W_LENGTH] = .DESCRIP [DSC$W_LENGTH];
```

```
;  1178          1545  5              TEMP_DSC [DSC$A_POINTER] = TEMP_BUF [0];
;  1179          1546  5              TEMP_DSC [DSC$B_SCALE] = .DESCRIP [DSC$B_SCALE];
;  1180          1547  5              BAS$$VA_STORE (.DESCRIP, .VALUE_LOCATION, TEMP_DSC)
;  1181          1548  5              END
;  1182          1549  4          ELSE
;  1183          1550  4              BAS$$VA_STORE (.DESCRIP, .VALUE_LOCATION, TEMP_BUF [0]);
;  1184          1551  4
;  1185          1552  4          END;
;  1186          1553  3
;  1187          1554  2      END;                                        ! end of INCR loop
;  1188          1555  2
;  1189          1556  2      !+
;  1190          1557  2      ! Update the number of elements in element 0 of the array.
;  1191          1558  2      !-
;  1192          1559  2
;  1193          1560  2      BEGIN
;  1194          1561  2      LOCAL
;  1195          1562  2          STR_LEN_LONG,
;  1196          1563  2          PTR;
;  1197          1564  2
;  1198          1565  2      STR_LEN_LONG = .STR_DESC [DSC$W_LENGTH];
;  1199          1566  2
;  1200          1567  2      IF (.DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA)
;  1201          1568  2      THEN
;  1202          1569  2          PTR = TEMP_BUF
;  1203          1570  2      ELSE
;  1204          1571  2          PTR = .DESCRIP [DSC$A_POINTER];
;  1205          1572  2
;  1206          1573  2      CASE .DESCRIP [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  1207          1574  2          SET
;  1208          1575  2
;  1209          1576  3          [DSC$K_DTYPE_B] :
;  1210          1577  4              BEGIN
;  1211          1578  4
;  1212          1579  4              IF .STR_LEN_LONG GTR 255
;  1213          1580  4              THEN
;  1214          1581  4                  BAS$$STOP(BAS$K_INTERR);
;  1215          1582  4
;  1216          1583  4              BLOCK [.PTR, 0, 0, %BPUNIT, 1] = .STR_DESC [DSC$W_LENGTH];
;  1217          1584  4
;  1218          1585  3              END;
;  1219          1586  3          [DSC$K_DTYPE_W] :
;  1220          1587  3              BLOCK [.PTR, 0, 0, %BPVAL/2, 1] = .STR_DESC [DSC$W_LENGTH];
;  1221          1588  3
;  1222          1589  3          [DSC$K_DTYPE_L] :
;  1223          1590  3              .PTR = .STR_DESC [DSC$W_LENGTH];
;  1224          1591  3
;  1225          1592  3          [DSC$K_DTYPE_F] :
;  1226          1593  3              CVTLF (STR_LEN_LONG, .PTR);
;  1227          1594  3
;  1228          1595  3          [DSC$K_DTYPE_D] :
;  1229          1596  4              BEGIN
;  1230          1597  4              !+
;  1231          1598  4              ! Apply scale even to this.
;  1232          1599  4              !-
;  1233          1600  4              LOCAL
;  1234          1601  4                  TEMP_DBL : VECTOR [2];
```

```
;  1235          1602  4              REGISTER
;  1236          1603  4                  R0 = 0,
;  1237          1604  4                  R1 = 1;
;  1238          1605  4              CVTLD (STR_LEN_LONG, TEMP_DBL);
;  1239          1606  4              BAS$SCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  1240          1607  4              TEMP_DBL [0] = .R0;
;  1241          1608  4              TEMP_DBL [1] = .R1;
;  1242          1609  4              BAS$COPY_D_R1 (TEMP_DBL [0], .PTR);
;  1243          1610  4              END;
;  1244          1611  3
;  1245          1612  3          [DSC$K_DTYPE_G] :
;  1246          1613  4              CVTLG (STR_LEN_LONG, .PTR);
;  1247          1614  3
;  1248          1615  3          [DSC$K_DTYPE_H] :
;  1249          1616  4              CVTLH (STR_LEN_LONG, .PTR);
;  1250          1617  3
;  1251          1618  3          [DSC$K_DTYPE_P] :
;  1252          1619  4              BEGIN
;  1253          1620  4              LOCAL
;  1254          1621  4                  TEMP_P : VECTOR [6,BYTE];
;  1255          1622  4
;  1256          1623  4              CVTLP (STR_LEN_LONG, %REF(10), TEMP_P);
;  1257          1624  4              ASHP (%REF(-.VALUE_DESCR [DSC$B_SCALE]), %REF(10), TEMP_P,
;  1258          1625  4                  %REF(0), VALUE_DESCR [DSC$W_LENGTH], .PTR);
;  1259          1626  4              END;
;  1260          1627  3
;  1261          1628  3          [DSC$K_DTYPE_DSC] :
;  1262          1629  4              BEGIN
;  1263          1630  4              LOCAL
;  1264          1631  4                  ELEM_DESC : REF BLOCK [8,BYTE];
;  1265          1632  4
;  1266          1633  4              IF .DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA
;  1267          1634  4              THEN
;  1268          1635  4                  BAS$$STOP (BAS$K_NOTIMP);       ! no virtual dyn mapped arrays
;  1269          1636  4
;  1270          1637  4              ELEM_DESC = .DESCRIP [DSC$A_POINTER];
;  1271          1638  4              CASE .ELEM_DESC [DSC$B_DTYPE] FROM DSC$K_DTYPE_B TO DSC$K_DTYPE_H OF
;  1272          1639  4              SET
;  1273          1640  4                  [DSC$K_DTYPE_B] :
;  1274          1641  5                      BEGIN
;  1275          1642  5
;  1276          1643  5                      IF .STR_LEN_LONG GTR 255
;  1277          1644  5                      THEN
;  1278          1645  5                          BAS$$STOP(BAS$K_INTERR);
;  1279          1646  5
;  1280          1647  5                      BLOCK [.ELEM_DESC [DSC$A_POINTER], 0, 0, %BPUNIT, 1]
;  1281          1648  5                          = .STR_DESC [DSC$W_LENGTH];
;  1282          1649  5
;  1283          1650  4                      END;
;  1284          1651  4                  [DSC$K_DTYPE_W] :
;  1285          1652  4                      BLOCK [.ELEM_DESC [DSC$A_POINTER], 0, 0, %BPVAL/2, 1]
;  1286          1653  4                          = .STR_DESC [DSC$W_LENGTH];
;  1287          1654  4
;  1288          1655  4                  [DSC$K_DTYPE_L] :
;  1289          1656  4                      .ELEM_DESC [DSC$A_POINTER] = .STR_DESC [DSC$W_LENGTH];
;  1290          1657  4
;  1291          1658  4                  [DSC$K_DTYPE_F] :                       ! 32-bit floating point
```

```
;  1292          1659  4                    CVTLF (STR_LEN_LONG, .ELEM_DESC [DSC$A_POINTER]);
;  1293          1660  4
;  1294          1661  4                [DSC$K_DTYPE_D] :                       ! 64-bit double floating
;  1295          1662  5                    BEGIN
;  1296          1663  5                    !+
;  1297          1664  5                    ! Apply scale to double value.
;  1298          1665  5                    !-
;  1299          1666  5                    LOCAL
;  1300          1667  5                        TEMP_DBL : VECTOR [2];
;  1301          1668  5                    REGISTER
;  1302          1669  5                        R0 = 0,
;  1303          1670  5                        R1 = 1;
;  1304          1671  5                    CVTLD (STR_LEN_LONG, TEMP_DBL);
;  1305          1672  5                    BAS$SCALE_D_R1 (.TEMP_DBL [0], .TEMP_DBL [1]);
;  1306          1673  5                    TEMP_DBL [0] = .R0;
;  1307          1674  5                    TEMP_DBL [1] = .R1;
;  1308          1675  5                    BAS$$COPY_D_R1 (TEMP_DBL [0], .ELEM_DESC [DSC$A_POINTER]);
;  1309          1676  4                    END;
;  1310          1677  4
;  1311          1678  4                [DSC$K_DTYPE_G] :                       ! G floating
;  1312          1679  4                    CVTLG (STR_LEN_LONG, .ELEM_DESC [DSC$A_POINTER]);
;  1313          1680  4
;  1314          1681  4                [DSC$K_DTYPE_H] :                       ! H floating
;  1315          1682  4                    CVTLH (STR_LEN_LONG, .ELEM_DESC [DSC$A_POINTER]);
;  1316          1683  4
;  1317          1684  4                [DSC$K_DTYPE_P] :                       ! decimal
;  1318          1685  5                    BEGIN
;  1319          1686  5                    LOCAL
;  1320          1687  5                        TEMP_P : VECTOR [6, BYTE];
;  1321          1688  5
;  1322          1689  5                        CVTLP (STR_LEN_LONG, %REF(10), TEMP_P);
;  1323          1690  5                        ASHP (%REF(-.ELEM_DESC [DSC$B_SCALE]), %REF(10),
;  1324          1691  5                            TEMP_P, %REF(0), ELEM_DESC [DSC$W_LENGTH],
;  1325          1692  5                            .ELEM_DESC [DSC$A_POINTER]);
;  1326          1693  4                        END;
;  1327          1694  4
;  1328          1695  4                    [INRANGE, OUTRANGE] :
;  1329          1696  4                        BAS$$STOP (BAS$K_DATTYPERR);
;  1330          1697  4
;  1331          1698  4                    TES;
;  1332          1699  4                    END;
;  1333          1700
;  1334          1701              [INRANGE, OUTRANGE] :
;  1335          1702                  BAS$$STOP (BAS$K_DATTYPERR);
;  1336          1703
;  1337          1704              TES;
;  1338          1705
;  1339          1706          IF .INDEX_ERROR GTR 0
;  1340          1707          THEN
;  1341          1708              BAS$$STOP (BAS$K_SUBOUTRAN);
;  1342          1709
;  1343          1710  4       IF (.DESCRIP [DSC$B_CLASS] EQL DSC$K_CLASS_BFA)
;  1344          1711  4       THEN
;  1345          1712  5           IF .DESCRIP [DSC$B_DTYPE] EQL DSC$K_DTYPE_P
;  1346          1713  5           THEN
;  1347          1714  4               BEGIN
;  1348          1715  4               LOCAL
```

```
;        4                   TEMP_DSC : BLOCK [12, BYTE];
;        4               TEMP_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_P;
;        4               TEMP_DSC [DSC$B_CLASS] = DSC$K_CLASS_SD;
;        4               TEMP_DSC [DSC$W_LENGTH] = .DESCRIP [DSC$W_LENGTH];
;        4               TEMP_DSC [DSC$A_POINTER] = .PTR;
;        4               TEMP_DSC [DSC$B_SCALE] = .DESCRIP [DSC$B_SCALE];
;        4               BAS$$VA_STORE (.DESCRIP, 0, TEMP_DSC)
;        4               END
;        3           ELSE
;        3               BAS$$VA_STORE (.DESCRIP, 0, .PTR);
;        3
;        2       END;
;        2
;        1       END;                                        ! end of STORE
; INFO#250              L1:1441
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250              L1:1442
; Referenced REGISTER symbol R1 is probably not initialized
; INFO#250              L1:1498
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250              L1:1499
; Referenced REGISTER symbol R1 is probably not initialized
; INFO#250              L1:1607
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250              L1:1608
; Referenced REGISTER symbol R1 is probably not initialized
; INFO#250              L1:1673
; Referenced REGISTER symbol R0 is probably not initialized
; INFO#250              L1:1674
; Referenced REGISTER symbol R1 is probably not initialized



                                    0FFC 00000 STORE:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11       ; 1203
                  5E       B0    AE   9E 00002         MOVAB   -80(SP), SP                                ;
                                 7E   D4 00006         CLRL    INDEX_ERROR                                ; 1237
    50      04    AC             04   C1 00008         ADDL3   #4, STR_DESC, R0                           ; 1275
                                 60   DD 0000D         PUSHL   (R0)                                       ;
                  58       08    AC   D0 0000F         MOVL    DESCRIP, R8                                ; 1281
    05      0A    A8             06   E1 00013         BBC     #6, 10(R8), 1$                             ;
                           0A    A8   95 00018         TSTB    10(R8)                                     ;
                                 08   19 0001B         BLSS    2$                                         ;
                  7E      00G    8F   9A 0001D 1$:     MOVZBL  #BAS$K_ARGDONMAT, -(SP)                    ;
       00000000G  00             01   FB 00021         CALLS   #1, BAS$$STOP                              ;
                  57       14    A8   9E 00028 2$:     MOVAB   20(R8), MULTIPLIERS                        ; 1283
                  50       0B    A8   9A 0002C         MOVZBL  11(R8), R0                                 ; 1284
            10    AE       14  A840   DE 00030         MOVAL   20(R8)[R0], BOUNDS                         ;
    0C      0A    A8             05   E1 00036         BBC     #5, 10(R8), 3$                             ; 1290
                  51             50   D0 0003B         MOVL    R0, LOW_INDEX                              ; 1293
                  50             01   D0 0003E         MOVL    #1, HIGH_INDEX                             ; 1294
            18    AE             01   CE 00041         MNEGL   #1, INDEX_INCR                             ; 1295
                                 07   11 00045         BRB     4$                                         ; 1290
                  51             01   D0 00047 3$:     MOVL    #1, LOW_INDEX                              ; 1299
            18    AE             01   D0 0004A         MOVL    #1, INDEX_INCR                             ; 1301
    16            06       02    A8   8F 0004E 4$:     CASEB   2(R8), #6, #22                             ; 1311
```

```
  002E      002E      002E      002E    00053 5$:    .WORD   6$-5$,-                        ;
  002E      002E      002E      002E    0005B                6$-5$,-                        ;
  002E      002E      002E      002E    00063                6$-5$,-                        ;
  0033      002E      002E      002E    0006B                6$-5$,-                        ;
  002E      002E      002E      002E    00073                6$-5$,-                        ;
            002E      002E      002E    0007B                6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             7$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$,-                        ;
                                                             6$-5$                          ;
                      56          68  3C 00081 6$:    MOVZWL  (R8), LENGTH                   ; 1328
                                  29  11 00084        BRB     10$                            ;
                      52          68  3C 00086 7$:    MOVZWL  (R8), R2                       ; 1316
                      52          02  C6 00089        DIVL2   #2, R2                         ;
                      56      01  A2  9E 0008C        MOVAB   1(R2), LENGTH                  ;
                 BF   8F      03  A8  91 00090        CMPB    3(R8), #191                    ; 1317
                                  18  12 00095        BNEQ    10$                            ;
                      52          01  D0 00097        MOVL    #1, I                          ; 1322
        53            01          52  78 0009A 8$:    ASHL    I, #1, R3                      ;
                      53          56  D1 0009E        CMPL    LENGTH, R3                     ;
                                  05  18 000A1        BGEQ    9$                             ;
                      56          53  D0 000A3        MOVL    R3, LENGTH                     ; 1323
                                  07  11 000A6        BRB     10$                            ;
        EE            52          09  F3 000A8 9$:    AOBLEQ  #9, I, 8$                      ; 1322
                      56          01  CE 000AC        MNEGL   #1, LENGTH                     ; 1321
                 2C   AE      04  BC  3C 000AF 10$:   MOVZWL  @STR_DESC, 44(SP)              ; 1336
   28   AE            51      18  AE  C3 000B4        SUBL3   INDEX_INCR, LOW_INDEX, 40(SP)  ; 1340
                 24   AE      18 BE40 9E 000BA        MOVAB   @INDEX_INCR[HIGH_INDEX], 36(SP) ; 1344
                              0C  AE  D4 000C0        CLRL    INDEX                          ; 1356
                                  42  11 000C3        BRB     14$                            ;
        50       0C   AE          01  C3 000C5 11$:   SUBL3   #1, INDEX, R0                  ; 1339
                      50          6E  C0 000CA        ADDL2   STR_BUF, R0                    ;
                 08   AE          60  9A 000CD        MOVZBL  (R0), STR_BUF_LONG             ;
                      55      28  AE  D0 000D1        MOVL    40(SP), INDEX_NUMBER           ; 1340
                 1C   AE      0C  AE  D0 000D5        MOVL    INDEX, INDEX_VALUE             ; 1341
                                  54  D4 000DA        CLRL    VALUE_LOCATION                 ; 1342
                      55      18  AE  C0 000DC 12$:   ADDL2   INDEX_INCR, INDEX_NUMBER       ; 1344
                 24   AE          55  D1 000E0        CMPL    INDEX_NUMBER, 36(SP)           ;
                                  34  13 000E4        BEQL    16$                            ;
        50            55          01  78 000E6        ASHL    #1, INDEX_NUMBER, R0           ; 1346
        51       10   AE          08  C3 000EA        SUBL3   #8, BOUNDS, R1                 ;
                    6140      1C  AE  D1 000EF        CMPL    INDEX_VALUE, (R1)[R0]          ;
                                  0C  19 000F4        BLSS    13$                            ;
        51       10   AE          04  C3 000F6        SUBL3   #4, BOUNDS, R1                 ; 1347
```

```
                 6140     1C   AE  D1 000FB          CMPL    INDEX_VALUE, (R1)[R0]                    ;
                               08  15 00100          BLEQ    15$                                      ;
            04    AE      0C   AE  D0 00102 13$:     MOVL    INDEX, INDEX_ERROR                       ; 1351
                             01AB  31 00107 14$:     BRW     39$                                      ; 1352
       50         54      FC A745  C5 0010A 15$:     MULL3   -4(MULTIPLIERS)[INDEX_NUMBER], -         ; 1356
                                                             VALUE_LOCATION, R0                       ;
       54         50      1C   AE  C1 00110          ADDL3   INDEX_VALUE, R0, VALUE_LOCATION          ;
                          1C   AE  D4 00115          CLRL    INDEX_VALUE                              ; 1357
                               C2  11 00118          BRB     12$                                      ; 1344
       50         54           56  C5 0011A 16$:     MULL3   LENGTH, VALUE_LOCATION, R0               ; 1360
       54         50      10   A8  C1 0011E          ADDL3   16(R8), R0, VALUE_LOCATION               ;
                          20   AE  D4 00123          CLRL    32(SP)                                   ; 1367
                  18      02   A8  91 00126          CMPB    2(R8), #24                               ;
                               31  12 0012A          BNEQ    19$                                      ;
                          20   AE  D6 0012C          INCL    32(SP)                                   ;
            4C    AE           64  B0 0012F          MOVW    (VALUE_LOCATION), VALUE_DESCR            ; 1374
            4E    AE      02   A4  90 00133          MOVB    2(VALUE_LOCATION), VALUE_DESCR+2         ; 1375
                  02      03   A4  91 00138          CMPB    3(VALUE_LOCATION), #2                    ; 1376
                               05  12 0013C          BNEQ    17$                                      ;
                  50           01  D0 0013E          MOVL    #1, R0                                   ;
                               04  11 00141          BRB     18$                                      ;
                  50      03   A4  9A 00143 17$:     MOVZBL  3(VALUE_LOCATION), R0                    ; 1377
            4F    AE           50  90 00147 18$:     MOVB    R0, VALUE_DESCR+3                        ; 1376
            50    AE      04   A4  D0 0014B          MOVL    4(VALUE_LOCATION), VALUE_DESCR+4         ; 1378
                  15      4E   AE  91 00150          CMPB    VALUE_DESCR+2, #21                       ; 1379
                               23  12 00154          BNEQ    20$                                      ;
            54    AE      08   A4  90 00156          MOVB    8(VALUE_LOCATION), VALUE_DESCR+8         ; 1384
                               1C  11 0015B          BRB     20$                                      ; 1367
            4C    AE           68  B0 0015D 19$:     MOVW    (R8), VALUE_DESCR                        ; 1389
            4E    AE      02   A8  90 00161          MOVB    2(R8), VALUE_DESCR+2                     ; 1390
            4F    AE           01  90 00166          MOVB    #1, VALUE_DESCR+3                        ; 1391
            50    AE           54  D0 0016A          MOVL    VALUE_LOCATION, VALUE_DESCR+4            ; 1392
                  15      4E   AE  91 0016E          CMPB    VALUE_DESCR+2, #21                       ; 1393
                               05  12 00172          BNEQ    20$                                      ;
            54    AE      08   A8  90 00174          MOVB    8(R8), VALUE_DESCR+8                     ; 1398
                          14   AE  D4 00179 20$:     CLRL    20(SP)                                   ; 1402
            BF    8F      03   A8  91 0017C          CMPB    3(R8), #191                              ;
                               08  12 00181          BNEQ    21$                                      ;
                          14   AE  D6 00183          INCL    20(SP)                                   ;
            50    AE      3C   AE  9E 00186          MOVAB   TEMP_BUF, VALUE_DESCR+4                  ; 1404
       16         06      02   A8  8F 0018B 21$:     CASEB   2(R8), #6, #22                           ; 1412
 0072  008D     0086          007F    00190 22$:     .WORD   27$-22$,-                                ;
 0072  0072     009B          0094    00198                  28$-22$,-                                ;
 0072  0072     0072          0072    001A0                  29$-22$,-                                ;
 00CE  0072     0072          0072    001A8                  26$-22$,-                                ;
 0072  0030     0072          0072    001B0                  30$-22$,-                                ;
       00C6     00BE          0072    001B8                  31$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             26$-22$,-                                ;
                                                             34$-22$,-                                ;
```

```
                                                                   26$-22$,-                         ;
                                                                   26$-22$,-                         ;
                                                                   23$-22$,-                         ;
                                                                   26$-22$,-                         ;
                                                                   26$-22$,-                         ;
                                                                   32$-22$,-                         ;
                                                                   33$-22$                           ;
                              42   11 001BE        BRB             26$                               ; 1527
             0B        14     AE   E9 001C0 23$:   BLBC            20(SP), 24$                       ; 1466
             7E        00G    8F   9A 001C4        MOVZBL          #BAS$K_NOTIMP, -(SP)              ; 1468
     00000000G  00            01   FB 001C8        CALLS           #1, BAS$$STOP                     ;
       16       06     4E     AE   8F 001CF 24$:   CASEB           VALUE_DESCR+2, #6, #22            ; 1470
002E   0049     0042      003B        001D4 25$:   .WORD           27$-25$,-                         ;
002E   002E     0057      0050        001DC                        28$-25$,-                         ;
002E   002E     002E      002E        001E4                        29$-25$,-                         ;
008A   002E     002E      002E        001EC                        26$-25$,-                         ;
002E   002E     002E      002E        001F4                        30$-25$,-                         ;
       0082     007A      002E        001FC                        31$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   34$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   26$-25$,-                         ;
                                                                   32$-25$,-                         ;
                                                                   33$-25$                           ;
             7E        00G    8F   9A 00202 26$:   MOVZBL          #BAS$K_DATTYPERR, -(SP)           ; 1521
     00000000G  00            01   FB 00206        CALLS           #1, BAS$$STOP                     ;
                              66   11 0020D        BRB             35$                               ;
          50    BE     08     AE   90 0020F 27$:   MOVB            STR_BUF_LONG, @VALUE_DESCR+4      ; 1474
                              5F   11 00214        BRB             35$                               ;
          50    BE     08     AE   B0 00216 28$:   MOVW            STR_BUF_LONG, @VALUE_DESCR+4      ; 1478
                              58   11 0021B        BRB             35$                               ;
          50    BE     08     AE   D0 0021D 29$:   MOVL            STR_BUF_LONG, @VALUE_DESCR+4      ; 1481
                              51   11 00222        BRB             35$                               ;
          50    BE     08     AE   4E 00224 30$:   CVTLF           STR_BUF_LONG, @VALUE_DESCR+4      ; 1484
                              4A   11 00229        BRB             35$                               ;
          34    AE     08     AE   6E 0022B 31$:   CVTLD           STR_BUF_LONG, TEMP_DBL            ; 1496
                50     34     AE   7D 00230        MOVQ            TEMP_DBL, R0                      ; 1497
                   00000000G  00   16 00234        JSB             BAS$SCALE_D_R1                    ;
          34    AE            50   7D 0023A        MOVQ            R0, TEMP_DBL                      ; 1498
                50     34     AE   9E 0023E        MOVAB           TEMP_DBL, R0                      ; 1500
                51     50     AE   D0 00242        MOVL            VALUE_DESCR+4, R1                 ;
                   00000000G  00   16 00246        JSB             BAS$$COPY_D_R1                    ;
                              27   11 0024C        BRB             35$                               ; 1470
          50    BE     08     AE4EFD  0024E 32$:   CVTLG           STR_BUF_LONG, @VALUE_DESCR+4      ; 1504
                              1F   11 00254        BRB             35$                               ;
          50    BE     08     AE6EFD  00256 33$:   CVTLH           STR_BUF_LONG, @VALUE_DESCR+4      ; 1507
```

```
                                 17  11 0025C        BRB     35$
          34  AE        0A   08  AE  F9 0025E 34$:   CVTLP   STR_BUF_LONG, #10, TEMP_P                   ; 1514
                        50   54  AE  98 00264        CVTBL   VALUE_DESCR+8, R0                           ; 1515
                        50       50  CE 00268        MNEGL   R0, R0
   00     34  AE        0A       50  F8 0026B        ASHP    R0, #10, TEMP_P, #0, VALUE_DESCR, -         ; 1517
                   50   BE   4C  AE     00271                @VALUE_DESCR+4
                        3C   14  AE  E9 00275 35$:   BLBC    20(SP), 39$                                 ; 1531
                        08   20  AE  E9 00279        BLBC    32(SP), 36$                                 ; 1535
                        7E  00G  8F  9A 0027D        MOVZBL  #BAS$K_NOTIMP, -(SP)
             00000000G  00       01  FB 00281        CALLS   #1, BAS$$STOP
                        15   02  A8  91 00288 36$:   CMPB    2(R8), #21                                  ; 1537
                                 19  12 0028C        BNEQ    37$
                   32   AE 0915  8F  B0 0028E        MOVW    #2325, TEMP_DSC+2                           ; 1542
                   30   AE       68  B0 00294        MOVW    (R8), TEMP_DSC                              ; 1544
                   34   AE   3C  AE  9E 00298        MOVAB   TEMP_BUF, TEMP_DSC+4                        ; 1545
                   38   AE   08  A8  90 0029D        MOVB    8(R8), TEMP_DSC+8                           ; 1546
                             30  AE  9F 002A2        PUSHAB  TEMP_DSC                                    ; 1547
                                 03  11 002A5        BRB     38$
                             3C  AE  9F 002A7 37$:   PUSHAB  TEMP_BUF                                    ; 1550
                                 54  DD 002AA 38$:   PUSHL   VALUE_LOCATION
                                 58  DD 002AC        PUSHL   R8
             00000000G  00       03  FB 002AE        CALLS   #3, BAS$$VA_STORE
 FE08     0C  AE        01   2C  AE  F1 002B5 39$:   ACBL    44(SP), #1, INDEX, 11$                      ; 1336
                        54   04  BC  3C 002BD        MOVZWL  @STR_DESC, STR_LEN_LONG                     ; 1565
                                 55  D4 002C1        CLRL    R5                                          ; 1567
                   BF   8F   03  A8  91 002C3        CMPB    3(R8), #191
                                 08  12 002C8        BNEQ    40$
                                 55  D6 002CA        INCL    R5
                        57   3C  AE  9E 002CC        MOVAB   TEMP_BUF, PTR                               ; 1569
                                 04  11 002D0        BRB     41$
                        57   04  A8  D0 002D2 40$:   MOVL    4(R8), PTR                                  ; 1571
              16        06   02  A8  8F 002D6 41$:   CASEB   2(R8), #6, #22                              ; 1573
 00E0       0051      004B      0031    002DB 42$:   .WORD   43$-42$,-
 00E0       00E0      005C      0057    002E3                45$-42$,-
 00E0       00E0      00E0      00E0    002EB                46$-42$,-
 0084       00E0      00E0      00E0    002F3                56$-42$,-
 00E0       009B      00E0      00E0    002FB                47$-42$,-
            007E      0078      00E0    00303                48$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             51$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             53$-42$,-
                                                             56$-42$,-
                                                             56$-42$,-
                                                             49$-42$,-
                                                             50$-42$
                              00AF   31 00309        BRW     56$                                         ; 1702
             000000FF   8F       54  D1 0030C 43$:   CMPL    STR_LEN_LONG, #255                          ; 1579
```

```
                                      0B   15 00313        BLEQ    44$
                    7E        00G     8F   9A 00315        MOVZBL  #BAS$K_INTERR, -(SP)                        ; 1581
          00000000G 00                01   FB 00319        CALLS   #1, BAS$$STOP
                    67        04      BC   90 00320 44$:   MOVB    @STR_DESC, (PTR)                            ; 1583
                                      4E   11 00324        BRB     52$                                         ; 1573
                    67        04      BC   B0 00326 45$:   MOVW    @STR_DESC, (PTR)                            ; 1587
                                      48   11 0032A        BRB     52$
                    67        04      BC   3C 0032C 46$:   MOVZWL  @STR_DESC, (PTR)                            ; 1590
                                      42   11 00330        BRB     52$
                    67                54   4E 00332 47$:   CVTLF   STR_LEN_LONG, (PTR)                         ; 1593
                                      3D   11 00335        BRB     52$
             34     AE                54   6E 00337 48$:   CVTLD   STR_LEN_LONG, TEMP_DBL                      ; 1605
                    50        34      AE   7D 0033B        MOVQ    TEMP_DBL, R0                                ; 1606
                    00000000G         00   16 0033F        JSB     BAS$SCALE_D_R1
             34     AE                50   7D 00345        MOVQ    R0, TEMP_DBL                                ; 1607
                    50        34      AE   9E 00349        MOVAB   TEMP_DBL, R0                                ; 1609
                    51                57   D0 0034D        MOVL    PTR, R1
                                    00BE   31 00350        BRW     65$
                    67            544EFD    00353 49$:     CVTLG   STR_LEN_LONG, (PTR)                         ; 1613
                                      6D   11 00357        BRB     57$
                    67            546EFD    00359 50$:     CVTLH   STR_LEN_LONG, (PTR)                         ; 1616
                                      67   11 0035D        BRB     57$
         34    AE   0A                54   F9 0035F 51$:   CVTLP   STR_LEN_LONG, #10, TEMP_P                   ; 1623
                    50        54      AE   98 00364        CVTBL   VALUE_DESCR+8, R0                           ; 1624
                    50                50   CE 00368        MNEGL   R0, R0
 00      34    AE   0A                50   F8 0036B        ASHP    R0, #10, TEMP_P, #0, VALUE_DESCR, (PTR)     ; 1625
                    67        4C      AE      00371
                                      7F   11 00374 52$:   BRB     63$                                         ; 1573
                    0B                55   E9 00376 53$:   BLBC    R5, 54$                                     ; 1633
                    7E        00G     8F   9A 00379        MOVZBL  #BAS$K_NOTIMP, -(SP)                        ; 1635
          00000000G 00                01   FB 0037D        CALLS   #1, BAS$$STOP
                    56        04      A8   D0 00384 54$:   MOVL    4(R8), ELEM_DESC                            ; 1637
               16   06        02      A6   8F 00388        CASEB   2(ELEM_DESC), #6, #22                       ; 1638
 002E          005D      0056        003B      0038D 55$:  .WORD   58$-55$,-
 002E          002E      006A        0064      00395               60$-55$,-
 002E          002E      002E        002E      0039D               61$-55$,-
 009A          002E      002E        002E      003A5               56$-55$,-
 002E          002E      002E        002E      003AD               62$-55$,-
               0093      008C        002E      003B5               64$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   68$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   56$-55$,-
                                                                   66$-55$,-
                                                                   67$-55$
                    7E        00G     8F   9A 003BB 56$:   MOVZBL  #BAS$K_DATTYPERR, -(SP)                     ; 1696
```

```
                    00000000G  00              01  FB 003BF          CALLS   #1, BAS$$STOP                                   ;
                                               74  11 003C6 57$:     BRB     69$                                             ;
                    000000FF   8F              54  D1 003C8 58$:     CMPL    STR_LEN_LONG, #255                              ; 1643
                                               0B  15 003CF          BLEQ    59$                                             ;
                               7E       00G    8F  9A 003D1          MOVZBL  #BAS$K_INTERR, -(SP)                            ; 1645
                    00000000G  00              01  FB 003D5          CALLS   #1, BAS$$STOP                                   ;
                          04   B6       04     BC  90 003DC 59$:     MOVB    @STR_DESC, @4(ELEM_DESC)                        ; 1648
                                               59  11 003E1          BRB     69$                                             ; 1638
                          04   B6       04     BC  B0 003E3 60$:     MOVW    @STR_DESC, @4(ELEM_DESC)                        ; 1653
                                               52  11 003E8          BRB     69$                                             ;
                          04   B6       04     BC  3C 003EA 61$:     MOVZWL  @STR_DESC, @4(ELEM_DESC)                        ; 1656
                                               4B  11 003EF          BRB     69$                                             ;
                          04   B6              54  4E 003F1 62$:     CVTLF   STR_LEN_LONG, @4(ELEM_DESC)                     ; 1659
                                               45  11 003F5 63$:     BRB     69$                                             ;
                          34   AE              54  6E 003F7 64$:     CVTLD   STR_LEN_LONG, TEMP_DBL                          ; 1671
                               50       34     AE  7D 003FB          MOVQ    TEMP_DBL, R0                                    ; 1672
                                  00000000G    00  16 003FF          JSB     BAS$$SCALE_D_R1                                 ;
                          34   AE              50  7D 00405          MOVQ    R0, TEMP_DBL                                    ; 1673
                               50       34     AE  9E 00409          MOVAB   TEMP_DBL, R0                                    ; 1675
                               51       04     A6  D0 0040D          MOVL    4(ELEM_DESC), R1                                ;
                                  00000000G    00  16 00411 65$:     JSB     BAS$$COPY_D_R1                                  ;
                                               23  11 00417          BRB     69$                                             ; 1638
                          04   B6          544EFD     00419 66$:     CVTLG   STR_LEN_LONG, @4(ELEM_DESC)                     ; 1679
                                               1C  11 0041E          BRB     69$                                             ;
                          04   B6          546EFD     00420 67$:     CVTLH   STR_LEN_LONG, @4(ELEM_DESC)                     ; 1682
                                               15  11 00425          BRB     69$                                             ;
          34   AE                       0A     54  F9 00427 68$:     CVTLP   STR_LEN_LONG, #10, TEMP_P                       ; 1689
                               50       08     A6  98 0042C          CVTBL   8(ELEM_DESC), R0                                ; 1690
                               50              50  CE 00430          MNEGL   R0, R0                                          ;
  00      34   AE                       0A     50  F8 00433          ASHP    R0, #10, TEMP_P, #0, (ELEM_DESC), -             ; 1692
                          04   B6              66     00439                  @4(ELEM_DESC)                                   ;
                                        04     AE  D5 0043C 69$:     TSTL    INDEX_ERROR                                     ; 1706
                                               0B  15 0043F          BLEQ    70$                                             ;
                               7E       00G    8F  9A 00441          MOVZBL  #BAS$K_SUBOUTRAN, -(SP)                         ; 1708
                    00000000G  00              01  FB 00445          CALLS   #1, BAS$$STOP                                   ;
                               2B              55  E9 0044C 70$:     BLBC    R5, 73$                                         ; 1710
                               15       02     A8  91 0044F          CMPB    2(R8), #21                                      ; 1712
                                               18  12 00453          BNEQ    71$                                             ;
                          32   AE     0915     8F  B0 00455          MOVW    #2325, TEMP_DSC+2                               ; 1717
                          30   AE              68  B0 0045B          MOVW    (R8), TEMP_DSC                                  ; 1719
                          34   AE              57  D0 0045F          MOVL    PTR, TEMP_DSC+4                                 ; 1720
                          38   AE       08     A8  90 00463          MOVB    8(R8), TEMP_DSC+8                               ; 1721
                                        30     AE  9F 00468          PUSHAB  TEMP_DSC                                        ; 1722
                                               02  11 0046B          BRB     72$                                             ;
                                               57  DD 0046D 71$:     PUSHL   PTR                                             ; 1725
                                               7E  D4 0046F 72$:     CLRL    -(SP)                                           ;
                                               58  DD 00471          PUSHL   R8                                              ;
                    00000000G  00              03  FB 00473          CALLS   #3, BAS$$VA_STORE                               ;
                                                   04 0047A 73$:     RET                                                     ; 1729

; Routine Size:  1147 bytes,    Routine Base:  _BAS$CODE + 05BF


; 1363          1730  1 END                                          ! end of module BAS$CHANGE
; 1364          1731  1
; 1365          1732  0 ELUDOM
```

```
;                                PSECT SUMMARY
;
;       Name                   Bytes                       Attributes
;
;   _BAS$CODE                      2618  NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)


;                          Library Statistics
;
;                                         -------- Symbols --------       Pages       Processing
;       File                              Total    Loaded   Percent       Mapped      Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1      9776       26         0          581        00:01.0


; Information:  16
; Warnings:     0
; Errors:       0


;                                 COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASCHANGE/OBJ=OBJ$:BASCHANGE MSRC$:BASCHANGE/UPDATE=(ENH$:BASCHANGE
;       )

; Size:         2618 code + 0 data bytes
; Run Time:        00:49.5
; Elapsed Time:    01:48.2
; Lines/CPU Min:     2100
; Lexemes/CPU-Min: 20371
; Memory Used:  351 pages
; Compilation Complete
```

BASCLOSE
LIS

BASCONCAT
LIS

BASCTRLO
LIS

BASCHANGE
LIS

BASCTRLC
LIS

BASCHAIN
LIS

BASCOPYFD
LIS

BASCHR
LIS

BASCMPAPP
LIS

BASCVTOUT
LIS

BASCOPOS
LIS